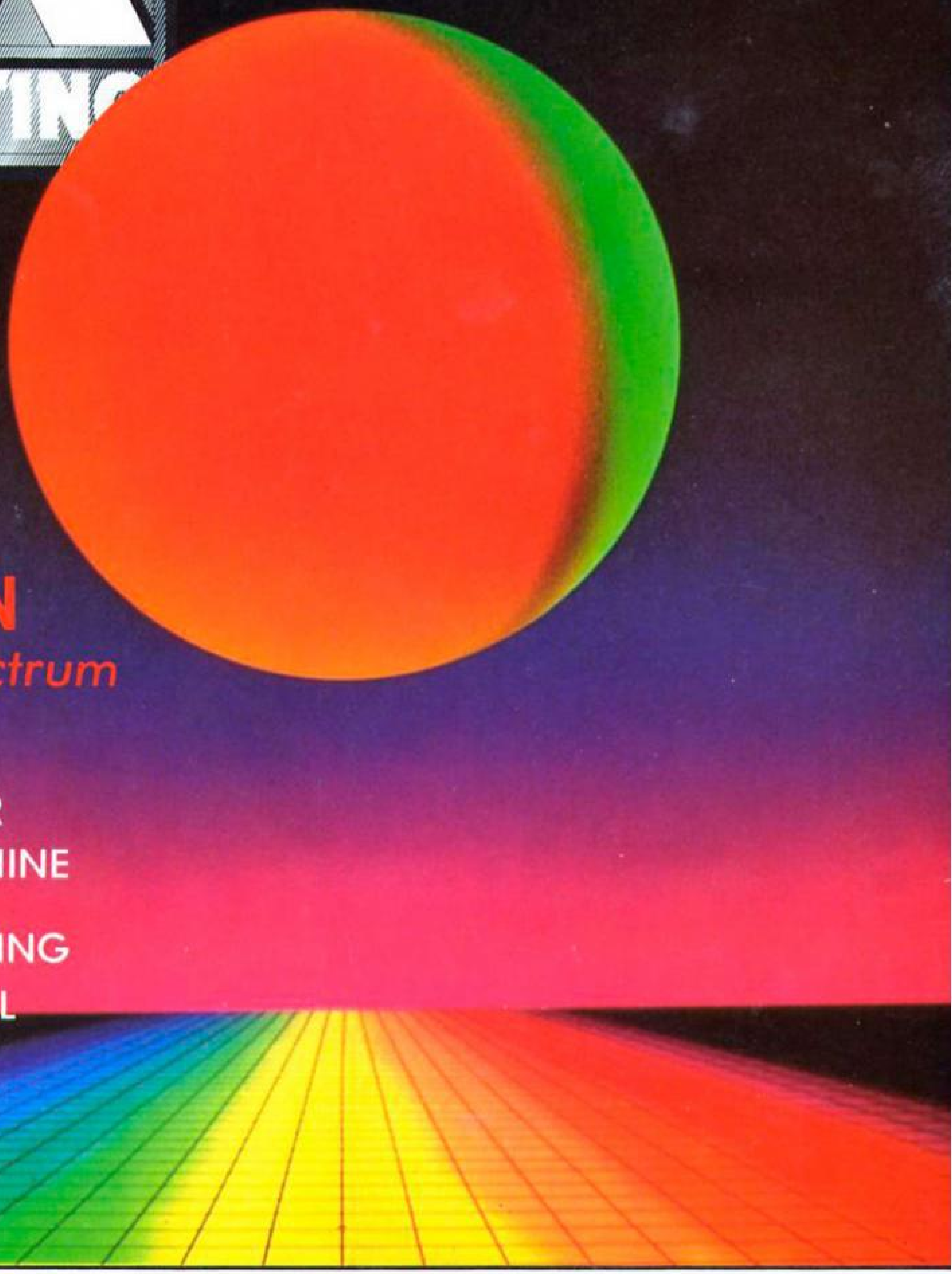
ZX COMPUTING MONTHLY

An Argus Specialist Publication NOVEMBER 1986 £1.50

FOR ALL SINCLAIR USERS



REVOLUTION
grips the Spectrum

SOUNDTEST:
SOUND SAMPLER
THE MUSIC MACHINE

QL ERROR TRAPPING
JOYSTICK SPECIAL

MONSTER HITS! ◆◆ Dan Dare ◆◆ The Trap Door ◆◆ Cyrus II

WIN! ◆ Saga Compliment ◆ Level 9 Yearplanners ◆ Gremlin Clocks


THANATOS
Spectrum in October, Amstrad in November, Commodore 64 in December.
All £8.95
DURELL
Sales dept.,
Castle Lodge, Castle Green,
Taunton, Somerset TA1 4AB
England
Telephone (0823) 54489 & 54029
software getting harder . . . .

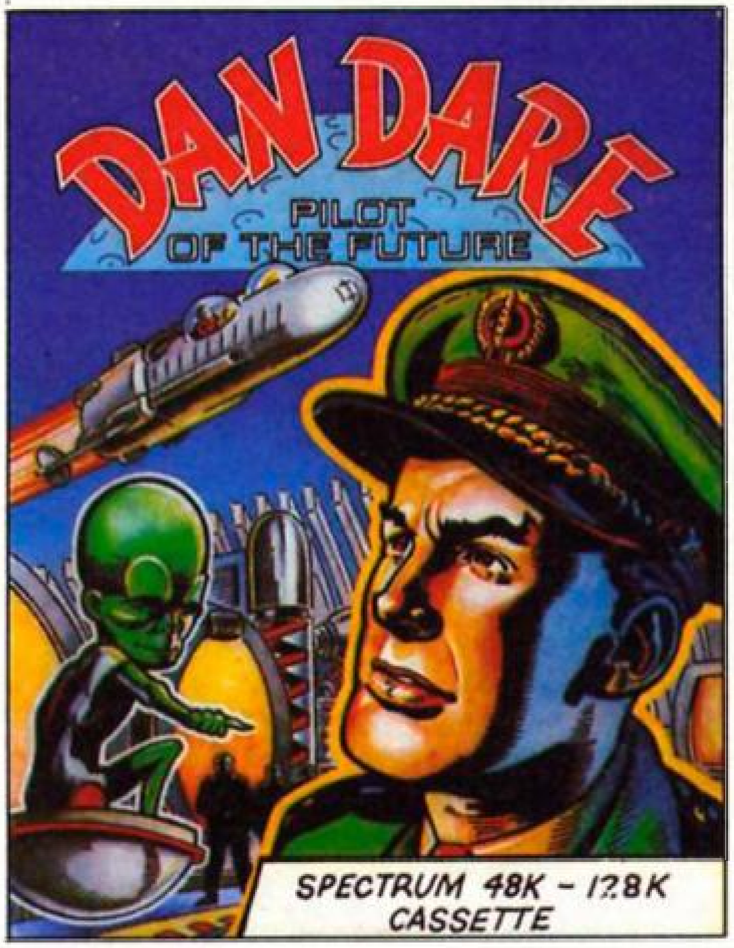
CONTENTS

FEATURES

REGULARS

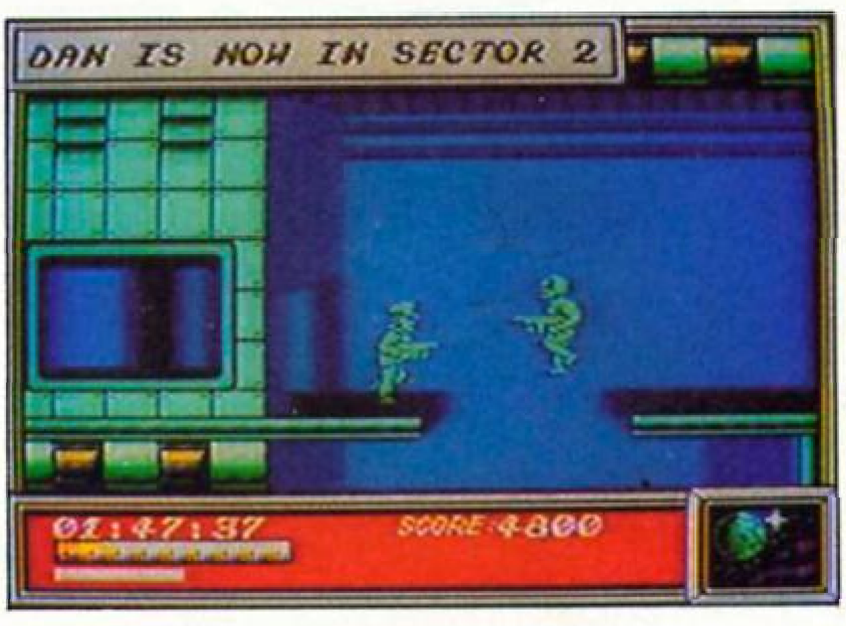
COMPETITIONS

SOFTWARE REVIEWS



Dan Dare reviewed (46)

PROGRAMMING PROJECTS



The Pilot of the Future in action (46)

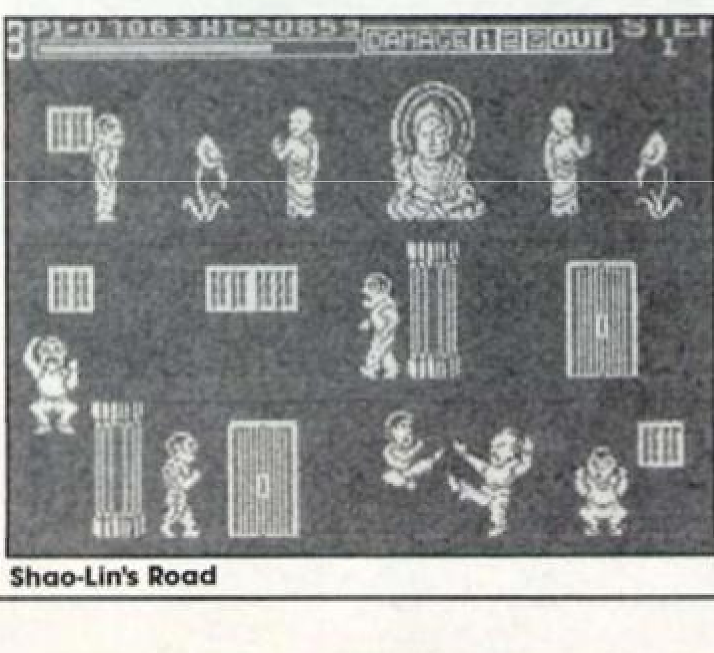
N·E·W·S

Fairlight II

Bo Jangeborg's follow up to Fairlight is now almost complete and due for an October release. Fairlight II carries straight on from where the action in Fairlight left off. But now, the game's hero, Isvar, has to leave Castle Fairlight and travel through a forest beset by the powers of Darkness. The Edge claim that this will be one of the first 'true' 128 games for the Spectrum and the 48K version will have to be loaded in two parts in order to fit into the smaller memory. Both versions will cost £9.95.

Also due from The Edge this month are Shao-Lin's Road, an arcade conversion of the sequel to Yie Ar Kung Fu!, and The Artist II. The latter is an upgraded version of the rather nifty Artist, which was also programmed by Bo Jangeborg, and includes many new features such as pull down menus, icons, AMX and Kempston mouse control and lots more. As with Fairlight II, The Artist II will be available in 48K and 128K versions for £14.95 and £17.95.





Shao-Lin's Road

The French Connection

Vera Cruz is an interesting new game from French company Infogrames. The aforementioned Vera is, or rather was, a well known woman of 'ow-you-say, easy virtue. Until she was discovered dead in her apartment — which is where you come in.

Using a police computer system to gather information you must piece together clues to solve the murder and apprehend the villain. But first you have to track the game down in your local software shop and bribe the owner with £9.95 before you can get on the trail.



Codemasters

This little bunch of Darlings (Jim, David and Richard) are collectively known as Code Masters, which is the name they've given to their new budget software label. Jim Darling used to work for AI Products who have provided many of Mastertronic's titles in the past, and his two sons have programmed a few games, such as the well known The Last V8, Masters of Magic and others.

Code Masters intends to release games as good as most full priced titles, but at a competitive £1.99. Their first Spectrum titles include Vampire ('platform game with a difference'), Phantomas, a graphic adventure called Necris Dome, and Terra-Incognita, the follow up to Mastertronic's odd game Nonterraqueous. Let's hope that those four get Code Masters off to a good start.



Opus meet the Plus 2

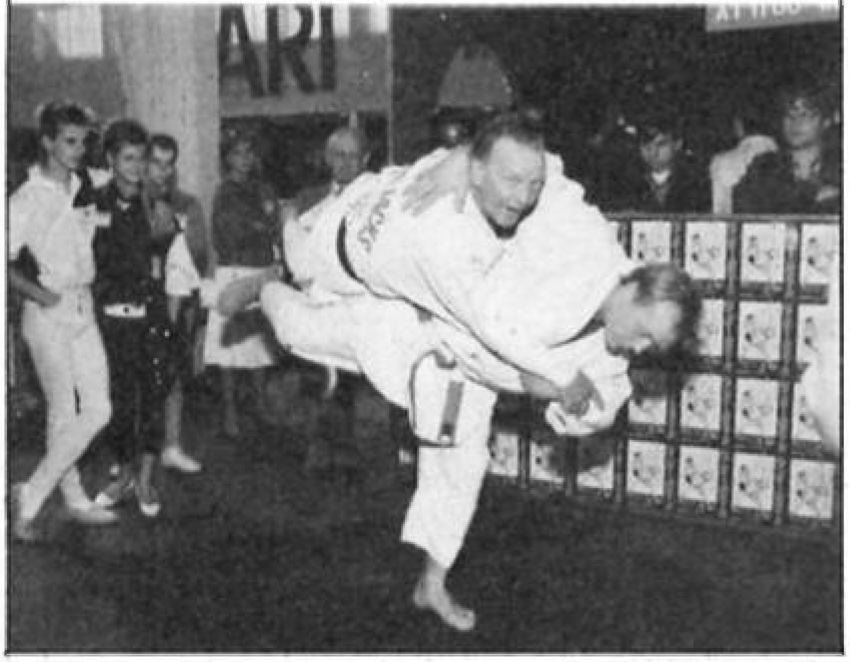
Opus Supplies wish to announce that yes, the Discovery disc drive is compatible with the new Spectrum Plus Two. Risking potential egg-on-face of mega proportions, Opus' John Harris tried out the new Spectrum when it went on show for the first time at the PCW show, and everything went fine. So Plus 2 owners looking for a disc system and not wanting to wait for the rumoured Amstrad system need look no further.

Discovery cost £99.95 plus VAT, and Opus can be contacted on 0737-65080.

Gratuitous violence

That's our only excuse for printing this picture of judo fiend Brian Jacks beating the daylights out of his poor old dad. That and the upcoming release of the judo simulation Uchi-Mata from Martech, on whose stand the two Jacks were demonstrating a few throws at last month's PCW show.

We would like to point out that ZX doesn't necessarily condone the mugging of one's father, though you have to admit that it's one way of getting him to go out and buy you a new Spectrum Plus 2.



Spectrum Games Top Ten

1	( )	Trivial Pursuit	Domark
2	( )	Dan Dare	Virgin
3	( )	Headcoach	Addictive
4	( )	Paperboy	Elite
5	( )	Dragons Lair	Software Projects
6	(2)	Ghosts 'n Goblins	Elite
7	( )	Heartland	Odin
8	( )	Xarq	Electric Dreams
9	(4)	Green Beret	Imagine
10	( )	Strike Force Harrier	Mirrorsoft

(Chart supplied by W.H. Smith)

All change at the top with some heavyweight releases this month. The question is can any game displace Trivial Pursuit before it comes into its own as the tailor made software stocking filler?

Strong contenders for the Christmas No 1 spot yet to be released include Gauntlet, (US Gold), Uridium (Hewson) and Trailblazer (Gremlin).

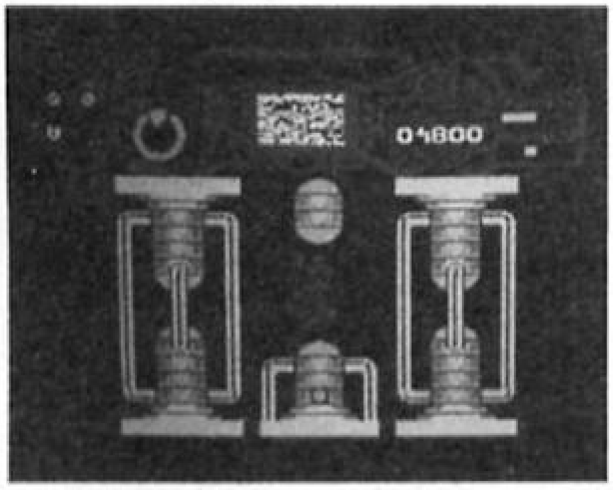
Now!

Virgin have just put together the third in their Now Games series. Now Games 3 is a compilation of five games, including a brand new Spectrum version of Sorcery which is available for the first time. The other titles on the tape are Nick Faldo Plays The Open, Codename Mat II, Everyone's A Wally, and the three part Bond Game, A View To A Kill. Now Games 3 should be in the shops as you read this, and costs £9.95.



At last, Antiriad

Despite a few last minute hitches, Palace Software's newie, The Sacred Armour of Antiriad should be released this month. This epic of barbarians and nuclear holocaust has already been previewed in ZX, so we will only add that the release date is 23rd October, and the price £8.99.





Dan Malone; Antiriad's author, hard at work on the graphics using a barbarian machine.

Editor: **Bryan Ralph**
Assistant Editor: **Cliff Joseph**
Consultant Editor: **Ray Elder**
Advertising Manager: **John McGarry**

Design: **Argus Design**
A.S.P. Advertising and Editorial
No. 1 Golden Square, London W1R 3AB 01-437-0626

Printed by Alabaster Passmore and Sons Ltd, Tovil, Maidstone, Kent.
Advertisement Copy Controler: Andy Selwood
Distributed by: Argus Press Sales and Distribution Ltd, 12-18 Paul Street, London EC2A 4JS

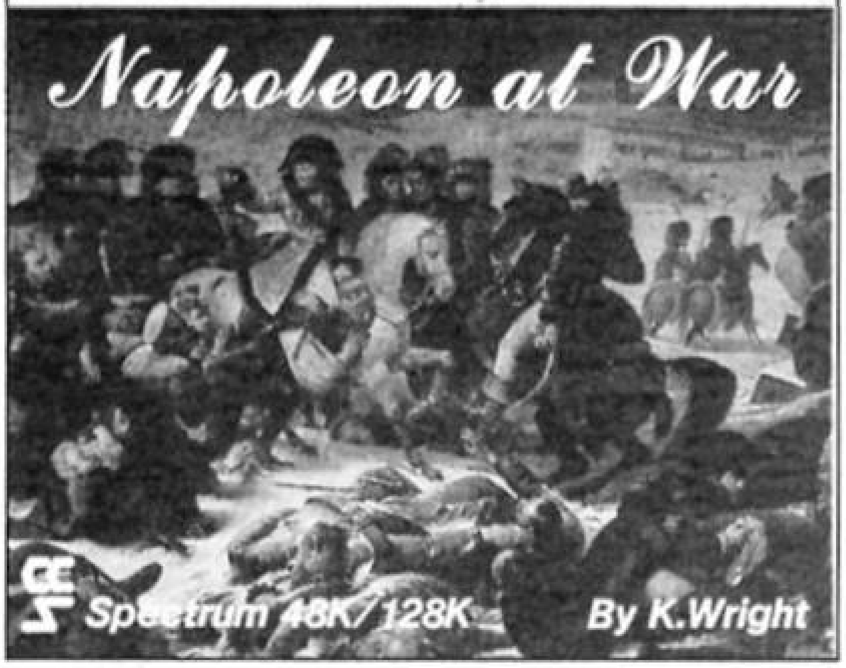
ZX Computing Monthly is published on the fourth Friday of each month. Subscription rates cn be obtained from ZX Subscriptions, Infonet, Times House, 179 The Marlowes, Hemel Hempstead, Herts HP51 1BB.

Not tonight Josephine. . .

Is a new strategy game in which Napoleon Bonaparte attempts to ——— no, let's try that again.

Napoleon At War is the latest release from wargames specialists CCS. Based upon the Battle of Eylau the game puts you in control of French troops as they battle the opposing (computer controlled) Russian and Prussian forces. CCS say that the game offers a high level of artificial intelligence, both in the computer opponent and in suggesting alternative tactics for your own forces, so if you're into wargames this is your chance to meet your Bakerloo for £8.95.

Further strategic info from CCS on 01-858-0763.



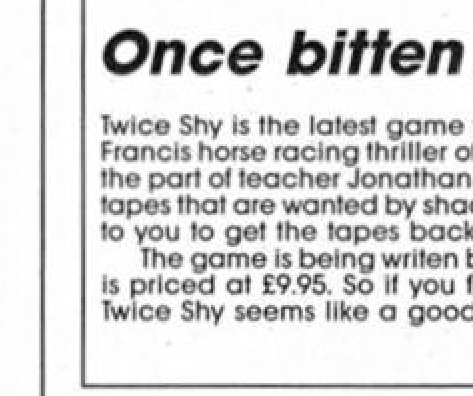
Once bitten. . .

Twice Shy is the latest game from Mosiac, based on the Dick Francis horse racing thriller of the same name. In it, you'll play the part of teacher Jonathan Derry who finds some computer tapes that are wanted by shady underworld figures, and it's up to you to get the tapes back to their rightful owner.

The game is being writen by The RamJam Corporation and is priced at £9.95. So if you fancy thrilling some race horses Twice Shy seems like a good bet.

Cruising

Microsphere, the software house that brought you Skool Daze, have been quiet of late but are about to surface again with a detective mystery game called Contact Sam Cruise. Sam Cruise is a private eye in the Bogart tradition with a permanently turned up collar. Microsphere are keeping shtumm about the plot thus far but here's a screenshot to get you into that 'film noir' mood.



Software Pirates

Ariolasoft offer you the chance to plan the crime of the century in their new game, They Stole A Million, in November. You are put in charge of a group of small-time villains and, with the help of some blueprints and your handpicked henchman, you can choose one of five 'targets' to rob and then you've got to find yourself a fence who can take the merchandise off your hands. If this sounds like your sort of thing, then beg, borrow or possibly steal £8.95 to get the goods from Ariolasoft.

Professional Retouch

Part of the listing in The Professional Touch (September) was printed out of sequence. The section of the listing C125 to C177 should be moved to the end of the listing.

Mantronix winners

The winner of our inlay design competition is 17 year old Ian Macaulay from Dudley, who sent in an impressive redesign of Probe Software's Mantronix cover. Ian's winning entry was very professionally presented and he gets the chance to see the professionals at work with a visit to Probe as they work on the theatrework for a new release.

Mr D. Nichols from Holborn, E. Bennet from London SW11, Jay Sachania, Milton Keynes and David Rademaker from Gent, Belgium win runners up prizes of a copy of Mantronix.



Elite Classic Winners

Five winners carry off the first ten releases in the Elite Classics range. They are Tommy Tay, Oberusel, West Germany, Jeffrey Ford, Etwall, Tony Parrit, Cleveland, Richard Haggis, London N3 and Cleasie Ward, Spilsby.

Graphic Adventure Creator Winners

ZX readers plumbed the depths of their imaginations to summon up the opening scenes of a new adventure and the results showed some fine sparks of originality. Ten winners get the chance of developing their ideas into a finished product with Incentive's Graphic Adventure Creator. They are Pete Pointon, Walsal; Jorgen Jacobsen, Glostrup, Denmark; A.F. Motin, Leeds; J.S. Russel, Fareham; Barry Carter, London E.2; D.M. Shannon, Co Antrim; W.J. Stewart, Nottingham; Emil Gruber, Graz, Austria; P. Newnham, Weston Super Mare; Roger Morton, Normanton.

Boggit Winners

Thirty ZX readers have booked their trip to the wacky land of enchantment, Muddle Earth by emerging the victors in our Boggit competition. They will each receive a copy of CRL's Monster Hit game and set off on this spoof adventure to discover the dreaded horrid squelchy things and other beings that Tolkien thought were too funny to include in the original.

The squad of winners are: H.H. Neilson, Galashields; P. Gaw, Normanton, Brett Coles, Milton Keynes; Miss E. Dron, Sutton; Trevor Boyd, Belfast; Geoffrey Priced, Pulborough; J. Riddell, Herne Bay; Richard Grant, West Lothian; Leif Kornstaedt, Colommieres, France; Javier Packer Coryn, Court St Ettienne, France; P.R. Ball, Brentwood; P.J. Nutter, Catterick; A.L. McLellan, Paisley; P.J.E. Williams, Fordingbridge; S. Burnett, Tottenham; Mrs P. Barclay, Plymouth; Gregory Teeger, London NW6; Colin MacKenzie, Fleet; D.J. Wagner, London W3; Sean Weekes, Sandiacre; Mark McEvilly, Birmingham; Phillip Hiscock, Shepton Mallet; John Jones, Lechlade; N.S. Collet, BFPO 34; N. Cashman, London E14; P. Harrison, Appleby; R. Record, Clacton; David Watson, Barnsley; Dirk Vanderreyken, Heverlee, Belgium; Skancmir Kais, Myslavokice, Poland.

COMPLIMENTS OF SAGA

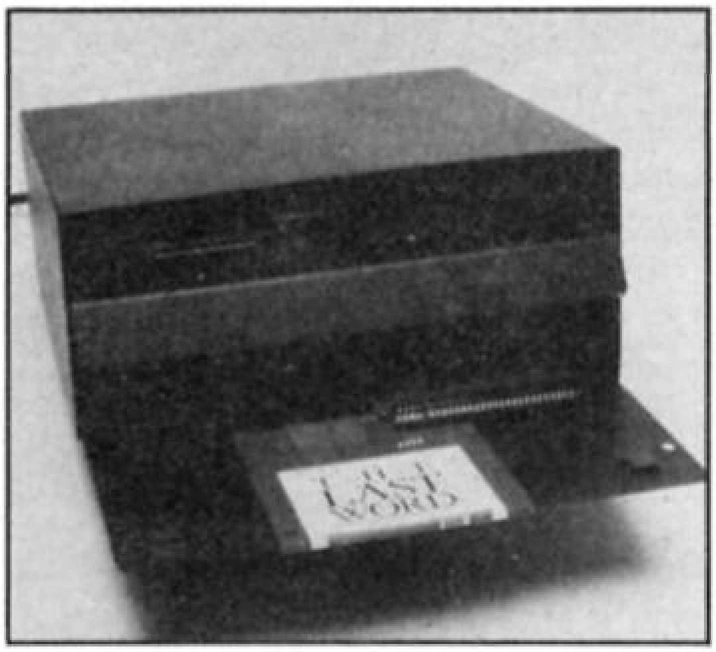
COMPETITION

Your chance to win a complete word processing set up from Saga worth over £300.

Saga Systems has put together a word processing package aimed at giving Spectrum owners a viable alternative to the temptations of the Amstrad PCW8256. Known as the Compliment it will be selling at £299+, that's £100 less than the Amstrad word processor. But the Compliment is made up of hardware that can be used for many other uses — it's far more than just a word processor.

The Prize

The Compliment consists of:
LQ Printer — prints 150 characters per second and 25 characters per second in LQM and also features an Epson compatible character set.
Parallel printer interface
Saga +2 Keyboard
Opus Discdrive
Last Word Software
Software Extention — includes foreign character sets
Mailmerge





Compliment Competition

I would like to win the Saga Compliment because (in 50 words or less)

....

....

....

....

....

Name: ....

Address: ....

....

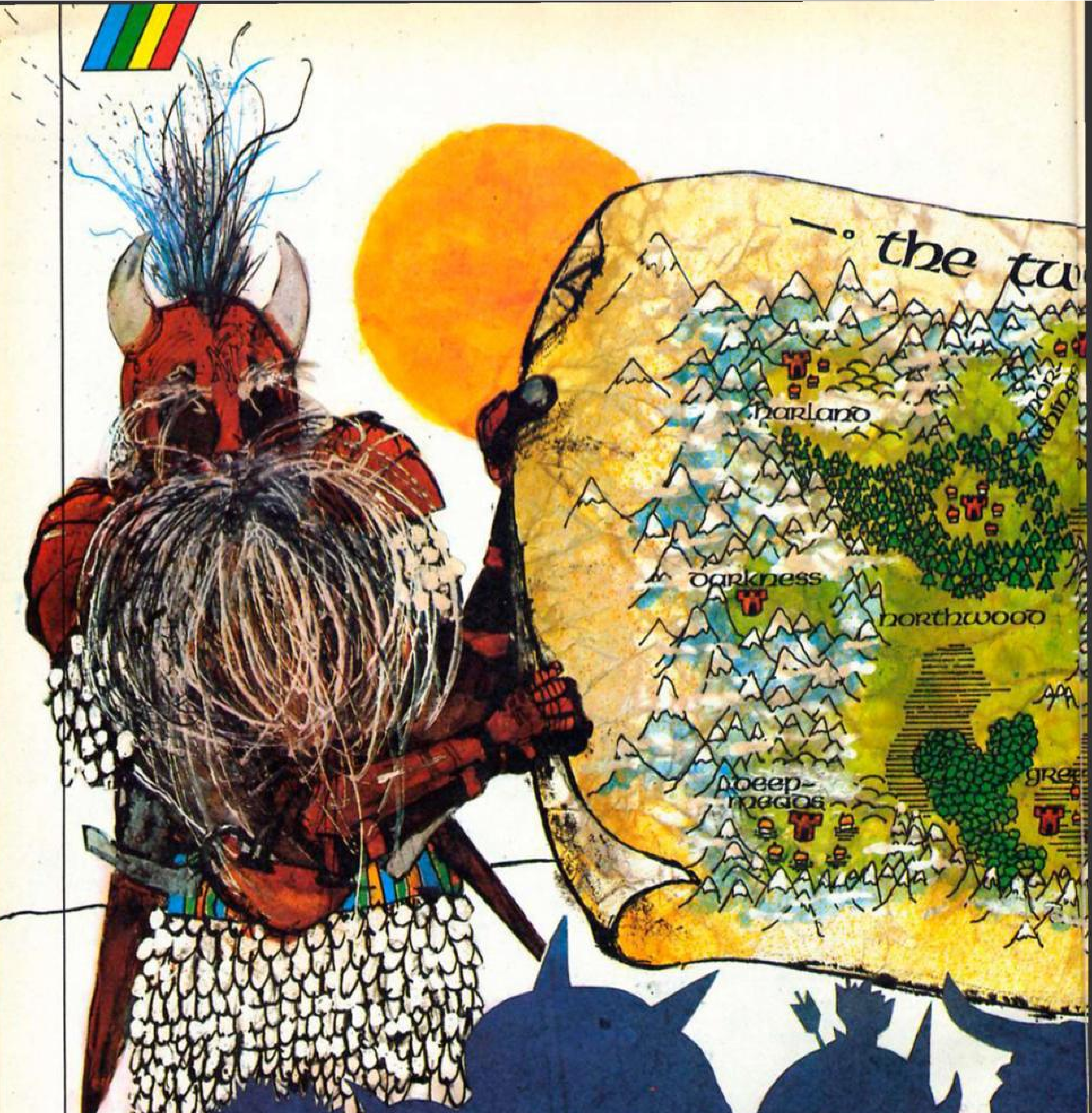
Post to Compliment Competition, ZX Computing Monthly, No 1 Golden Square, London W1R 3AB.

The competition

In conjunction with Saga, ZX is giving one of its readers the opportunity to win a Compliment system. All you have to do is give us in 50 words or less your reason for wanting to win the Saga Compliment.

The competition is open to all readers of ZX except employees of Argus Specialist publications, Alabaster Passmore and Saga Systems. The editor's decision is final and no correspondence can be entered into. The closing date is December 5th 1986.

Fill out the coupon below and post to Compliment Competition, ZX Computing Monthly, No 1 Golden Square, London W1R 3AB.



the war of

Alan Davis begins a three part series to create an original fantasy wargame.

Roland stood by the keep, watching the folk of Greenways gathering their crops from the nearby fields. It had been a fine summer — one of the best he could recall — and the Shirefolk were in good heart. There would be ample provision for Winter this year, and he was content.

Suddenly the excited shouts and gestures of the villagers drew his attention towards the forest of Greenways in the west, whence a solitary rider was approaching at great speed.

"What news, friend?" he called, as the newcomer dismounted and tethered his beast.

"Ill, my Lord," gasped the rider, gulping thankfully at a proffered draught of water. "I



the shires

come in all haste from the mountains of Darkness, where great armies are gathering in fearsome numbers!"

Roland shook his head.

"Nay, friend — we surely have little to fear from the Shire of Darkness. The mountains are impassable. No army could penetrate them and survive."

"My Lord, the forces of Darkness are mounted upon guruks — great beasts from the far west, capable of carrying many men over the most rugged terrain, at speed. Even water is no obstacle to them, for they are mighty swimmers. Believe me, my Lord, I have seen these things with my own eyes but three days ago, and even now Deepmeads and Northwood may have fallen!"

Roland needed no further persuasion.

"Then all the Shires are in the gravest peril! Freemen of Greenways, rally to my banner and prepare for war! We ride at once to warn Ranolf of Deepmeads. . ." And as Roland spoke, from the west came the faint but ominous sound of thunder. . . . .

▸

the war of the shires

Don't worry — you haven't bought a role-playing fantasy magazine by mistake! This really is ZXC Monthly, and what you've just read is the introduction to a computer game. I'd better explain...

It all began with a bit of doodling, really. With pen and paper to hand one morning, I started sketching out a map of an imaginary world, with nothing very particular in mind — not knowing where it might lead. Gradually the map took shape, dividing itself interestingly into twelve Shires. Each Shire acquired a keep and a sprinkling of villages, and of course these in turn required inhabitants. So I started inventing some characters — thanes of the Shires: Roland of Greenways, Ranolf of Deepmeads, and so on.

Now, someone (C.S. Lewis, I think) once commented that imaginary worlds are all very well, but when you've created this interesting place and populated it with characters, something has to *happen* in it! It was at about this stage that I began to realise that there were possibilities here for a computer game of the "fantasy-epic-wargame" variety, which might provide some interesting programming examples (as well as an entertaining game) for my friends the readers of ZXC — not to mention myself! So I discarded the pen and paper, plugged in the old Spectrum — and the eventual result was "THE WAR OF THE SHIRES".

Building a world

Right at the start, a number of decisions had to be made about the format of the game. At first I did briefly consider using a simplified form of landscaped graphics, but rejected the idea on the grounds that the sheer volume of graphics data would make the program listings ridiculously long. However, there seemed to be no reason why the original "world map" itself couldn't be represented graphically — so I settled for that. The next question concerned the actual programming, which, ideally, should be sufficiently flexible for others to modify and add developments of their own. I could see from the start that *some* machine code programming would be necessary to prevent ponderous response times, but I was surprised to discover that in practise only two machine code routines were strictly necessary, and that BASIC was perfectly adequate for everything else.

Now this is good news, because of course it's very much easier to experiment with changes in a BASIC program than it is with machine code. The final requirements was merely that the game should be fun to play in its own right, regardless of any further developments you might like to make yourself in due course.

So if, as I am, you're attracted by the idea of creating imaginary worlds; or if you have a taste for the "Lords of Midnight" type of epic fantasy game which can be played over and again in different ways; or if you just fancy tackling the interesting programming challenges of a strategy game — why not join me for the next couple of months, and try writing your own....?

LISTING 1
Map printing routine:
Assembler program.

```
*HISOFT GENS3M2 ASSEMBLER*
        ZX SPECTRUM

Copyright (C) HISOFT 1983,4
All rights reserved

Pass 1 errors: 00

                      10 *D+
         20 *C-
         30 PRMAP
60000    40              ORG   60000
60000    50              LD    A,2
60002    60              CALL  #1601
60005    70              CALL  #0D6B
60008    80              LD    A,2
60010    90              CALL  #1601
60013   100              LD    HL,(VARS)
60016   110              LD    DE,8
60019   120              ADD   HL,DE
60020   130              PUSH  HL
60021   140              LD    DE,704
60024   150              ADD   HL,DE
60025   160              EX    DE,HL
60026   170              POP   HL
60027   180 START        LD    B,32
60029   190 LOOP         LD    A,16
60031   200              RST   16
60032   210              LD    A,(HL)
60033   220              LD    (CHR),A
60036   230              CALL  COLOUR
60039   240              RST   16
60040   250              LD    A,(CHR)
60043   260              RST   16
60044   270              INC   HL
60045   280              DJNZ  LOOP
60047   290              LD    A,E
60048   300              CP    L
60049   310              JP    NZ,START
60052   320              LD    A,D
60053   330              CP    H
60054   340              JP    NZ,START
60057   350              RET
60058   360 COLOUR       CP    144
60060   370              JP    Z,BLUE
60063   380              CP    145
60065   390              JP    Z,GREEN
60068   400              CP    146
60070   410              JP    Z,CYAN
60073   420              CP    147
60075   430              JP    Z,GREEN
60078   440              CP    148
60080   450              JP    Z,RED
60083   460              CP    149
60085   470              JP    Z,MAGE
60088   480              CP    150
60090   490              JP    Z,GREEN
60093   500              CP    151
60095   510              JP    Z,GREEN
60098   520              XOR   A
60099   530              RET
60100   540 BLUE         LD    A,1
60102   550              RET
60103   560 RED          LD    A,2
60105   570              RET
60106   580 MAGE         LD    A,3
60108   590              RET
60109   600 GREEN        LD    A,4
60111   610              RET
60112   620 CYAN         LD    A,5
60114   630              RET
60115   640 CHR          DEFB  0
23627   650 VARS         EQU   23627

Pass 2 errors: 00

BLUE     EAC4      CHR      EAD3
COLOUR   EA9A      CYAN     EAD0
GREEN    EACD      LOOP     EA70
MAGE     EACA      PRMAP    87F4
RED      EAC7      START    EA78
VARS     5C4B

Table used:    137 from    194
```

Maps and things

This month, as you might hve guessed from the illustrations, is mostly about graphics and maps — specifically, the business of translating the original map of the Twelve Shires into something that can be readily programmed on the Spectrum. The Spectrum upper screen, as you know, consists of 22 rows of 32 character squares. Suppose we let each of these character squares represent one 'location' on the map of the world, then. This will give us 704 locations to play with — which should give ample scope for our "mini-epic", and will enable us to display the entire worldmap on screen at once.

But how should we store such a map in memory? One method would be to draw it out on-screen, and then store the entire screen as a code block above RAMTOP, calling it up as required during the game using a short LDIR-based machine code routine. Well, if the map were to remain unchanged during play, this would be fine — but in "The War of the Shires" I wanted certain features of the map to be altered by events. (Ships, for instance, won't always be in the same place, and keeps may be reduced to rubble in battle.) I finally decided to store the map in a BASIC character array, x$(22,32), so that each character square on the screen could be directly related to a single element of the array, for ease of manipulation. Mountains, keeps, and so on, could then be represented by UDGs on screen, with the codes for the UDGs being held in the array.

So how do we actually get the map onto the screen, with each UDG appearing in its correct colour? Well, it *could* be done in BASIC — but it's a painfully slow process which would drive any player of the game to distraction (in play, the game involves a good deal of swapping between the map and the various text "pages" dealing with battles and the like.) This, then, *must* be a machine code job — and the assembler program in Listing 1 will do it for us. It's a very simple routine which assumes that the array x$(22,32) is stored at the *beginning* of he BASIC variables area, and is simply prints the 704 characters contained in the array in the appropriate INK on white PAPER. I must emphasise that the routine won't work if the array x$() isn't at the start of the variables area. This means that either the array must be dimensioned before any other

LISTING 2
BASIC program to generate machine code block and udg's.

```
   1 REM ************************
        Machine code and UDGs
        ************************
  10 CLEAR 59999: LET s=0
  20 FOR i=60000 TO 60114
  30 READ x: LET s=s+x: POKE i,x
  40 NEXT i
  50 FOR i=60200 TO 60366
  60 READ x: LET s=s+x: POKE i,x
  70 NEXT i
  80 FOR i=65368 TO 65455
  90 READ x: LET s=s+x: POKE i,x
 100 NEXT i
 110 IF s<>46245 THEN PRINT AT 1
0,12;"ERROR!!!": STOP
 120 PRINT AT 10,7;"SAVING MACHI
NE CODE"
 130 SAVE "PRINT"CODE 60000,370
 140 CLS : PRINT AT 10,11;"SAVIN
G UDGS"
 150 SAVE "UDGS"CODE 65368,88
 200 DATA 62,2,205,1,22,205,107,
13,62,2
 201 DATA 205,1,22,42,75,92,17,0
,0,25
 202 DATA 229,17,192,2,25,235,22
5,6,32,62
 203 DATA 16,215,126,50,211,234,
205,154,234,215
 204 DATA 58,211,234,215,35,16,2
38,123,189,194
 205 DATA 123,234,122,188,194,12
3,234,201,254,144
 206 DATA 202,196,234,254,145,20
2,205,234,254,146
 207 DATA 202,208,234,254,147,20
2,205,234,254,148
 208 DATA 202,199,234,254,149,20
2,202,234,254,150
 209 DATA 202,205,234,254,151,20
2,205,234,175,201
 210 DATA 62,1,201,62,2,201,62,3
,201,62
 211 DATA 4,201,62,5,201
 300 DATA 42,75,92,126,254,90,20
2,56,235,205
 301 DATA 184,25,235,195,43,235,
35,126,50,203
 302 DATA 235,35,35,34,205,235,2
05,127,235,254
 303 DATA 1,202,181,235,205,146,
235,205,181,235
 304 DATA 42,205,235,237,91,204,
235,22,0,25
 305 DATA 34,205,235,58,204,235,
71,58,203,235
 306 DATA 144,50,203,235,42,205,
235,126,254,32
 307 DATA 194,66,235,35,34,205,2
35,58,203,235
 308 DATA 61,50,203,235,195,104,
235,58,203,235
 309 DATA 254,33,218,137,235,175
,201,58,203,235
 310 DATA 50,204,235,62,1,201,42
,205,235,17
 311 DATA 31,0,25,126,254,31,210
,165,235,19
 312 DATA 123,50,204,235,201,35,
126,254,32,194
 313 DATA 175,235,195,159,235,27
,43,43,195,153
 314 DATA 235,62,2,205,1,22,237,
91,205,235
 315 DATA 237,75,204,235,6,0,205
,60,32,62
 316 DATA 13,215,201,24,24,119,1
78
 400 DATA 16,24,60,52,118,114,24
2,105,0,0
 401 DATA 0,56,116,242,56,0,0,10
2,153,0
 402 DATA 0,102,153,0,0,48,108,5
0,110,124
 403 DATA 36,0,0,0,24,60,126,52,
52,0
 404 DATA 0,219,255,90,126,102,2
31,0,0,68
 405 DATA 0,40,2,0,80,0,0,16,56,
16
 406 DATA 56,124,16,0,22,28,60,1
24,254,16
 407 DATA 215,126,0,0,0,0,0,0,0,
0
 408 DATA 0,252,124,4,4,85,85,25
5
```

variables are assigned, or the variables area must first be CLEARed before loading the array from tape into a BASIC program.

Listing 2 will POKE in the necessary bytes for the map printing routine from BASIC — but it also contains two other important code blocks. The first of these "extras" is a machine code routine for printing text held in the BASIC variable z$, in such a way that words are not broken at the ends of lines. I haven't given the assembly language program for this, since it's very similar to a routine I gave in the May 1986 issue of ZXC. The only important difference is that this routine prints continuously downwards from the current PRINT position, whereas the previous routine produced a scrolling display. It's convenient to have both the map and text printing routines stored on tape as a single code block (saved as **"PRINT" CODE 60000,370**) — which is why I've given them together now even though we won't be using the text printing routine until next month. The call addresses, by the way, are USR 60000 and USR 60200 respectively.

UDGs

The last chunk of data in Listing 2 defines the UDGs required for the game.

The program will save these as separate code block (**"UDGS" CODE 65368,88**). You should note that the map printing routine will print the correct colours for the graphics characters only as I've defined them in figure 1.

Now for the moment of truth. Somehow, we need to get the data for 704 locations into our map array — and if I were to present you with a list of 704 numbers to type in, you wouldn't be very pleased, would you? Can't say I'd blame you! Let's make ourselves a map editing program instead. Type in Listing 3, save it to tape with **SAVE "MAP EDITOR" LINE 1**, and then stop the tape in that position and put it to one side. Enter **CLEAR 59999** as a direct command, and load in the two code blocks you saved from Listing 2 thus: **LOAD "PRINT" CODE: LOAD "UDGS" CODE**. Now replace the editor tape and **SAVE "PRINT" CODE 60000,370: SAVE "UDGS" CODE 65368,88**. (They must be saved in this order). Rewind the tape, type **LOAD ""**, and wait for the menu to appear.

At the top of the screen you'll see the various UDG shapes presented for ease of reference

Figure 1. Codes for the map's graphic characters.

CHR$ CODE	SHAPE	COLOUR
144	Mountain	Blue
145	Downs	Green
146	Water	Cyan
147	Woodland	Green
148	Village	Red
149	Keep	Magenta
150	Meadows	Green
151	Pine forest	Green
152	Ship	Black
153	(Plain)	—
154	Army	Black

There's no reason at all why you shouldn't change the shapes themselves using your favourite UDG designer, if you wish. But you mustn't, at this stage, change their *meaning*, or their order if you're to avoid getting into a terrible mess later on! CHR$ 144 (or GRAPHICS SHIFT/A,, if you prefer) *must* be a mountain, and so on.

— except that the "Army" graphic will be missing since this doesn't form a permanent part of the map. Ignore options 1 and 2 for the present, and select option 3: "Edit map". You'll be greeted by a blank screen containing a white cursor square, with brief instructions displayed on the bottom two lines. Try moving the cursor

LISTING 3
The map editor BASIC program.

```
   1 REM ************************
              MAP EDITOR
         ************************

   5 DEF FN i(x)=(x=144)+(2 AND
x=148)+(3 AND x=149)+(4 AND (x=1
45 OR x=147 OR x=150 OR x=151))+
(5 AND x=146)
  10 BORDER 6: CLEAR 59999: LOAD
 "PRINT"CODE : LOAD "UDGS"CODE :
 DIM x$(22,32)
  20 CLS : PRINT AT 10,11;"MAP M
AKER";AT 13,10;"1: Load map";AT
15,10;"2: Save map";AT 17,10;"3:
 Edit map"
  25 PRINT AT 1,14;"UDGS";AT 3,8
;"A B C D E F G H I";AT 4,8;: FO
R i=144 TO 152: PRINT INK FN i(i
);CHR$ i+" ";: NEXT i
  30 PRINT #1; BRIGHT 1;AT 0,9;"
SELECT OPTION": PAUSE 0: LET i$=
INKEY$: IF INKEY$<"1" OR INKEY$>
"3" THEN GO TO 30
  40 IF i$="1" THEN CLEAR : LOAD
 "MAP" DATA x$(): PRINT AT 10,4;
"Loading completed.": PAUSE 50:
GO TO 20
  50 IF i$="2" THEN CLS : SAVE "
MAP" DATA x$(): PRINT AT 10,4;"S
aving completed.": PAUSE 50: GO
TO 20
  60 LET x=15: LET y=10: RANDOMI
ZE USR 60000: GO SUB 1000
  80 INK 8: PAPER 8: PRINT BRIGH
T 1;AT y,x; OVER 1;" ": PAUSE 0:
 LET i$=INKEY$
  90 IF i$="q" THEN GO TO 20
 100 IF i$=CHR$ 13 THEN GO TO 20
0
 110 IF i$=CHR$ 12 THEN PRINT AT
 y,x;" ": LET x$(y+1,x+1)=" ": G
O TO 80
 120 PRINT BRIGHT 0;AT y,x; OVER
 1;" ": LET x=x-(i$="5" AND x>0)
+(i$="8" AND x<31): LET y=y-(i$=
"7" AND y>0)+(i$="6" AND y<21):
GO TO 80
 200 GO SUB 1010: PRINT BRIGHT 1
;AT y,x; OVER 1;" "
 210 PRINT AT y,x; FLASH 1; OVER
 1;" ": PAUSE 0: LET i$=INKEY$:
IF i$<"a" OR i$>"i" THEN GO TO 2
10
 220 LET i$=CHR$ (47+CODE i$): P
RINT AT y,x; INK FN i(CODE i$);
FLASH 0;i$: LET x$(y+1,x+1)=i$:
GO SUB 1000: GO TO 80
1000 PRINT #1; BRIGHT 1;AT 0,0;
PAPER 6;"5,6,7,8= Move cursor ";
 PAPER 7;"  q= Quit  ";"CS/0= De
lete "; PAPER 5;" ENTER= Add gra
phic": RETURN
1010 PRINT #1; BRIGHT 1;AT 0,0;"
Press the appropriate letter for
the graphic you wish to add here
": RETURN
```

the war of the shires

FIGURE 2
The on-screen map of the 12 shires.

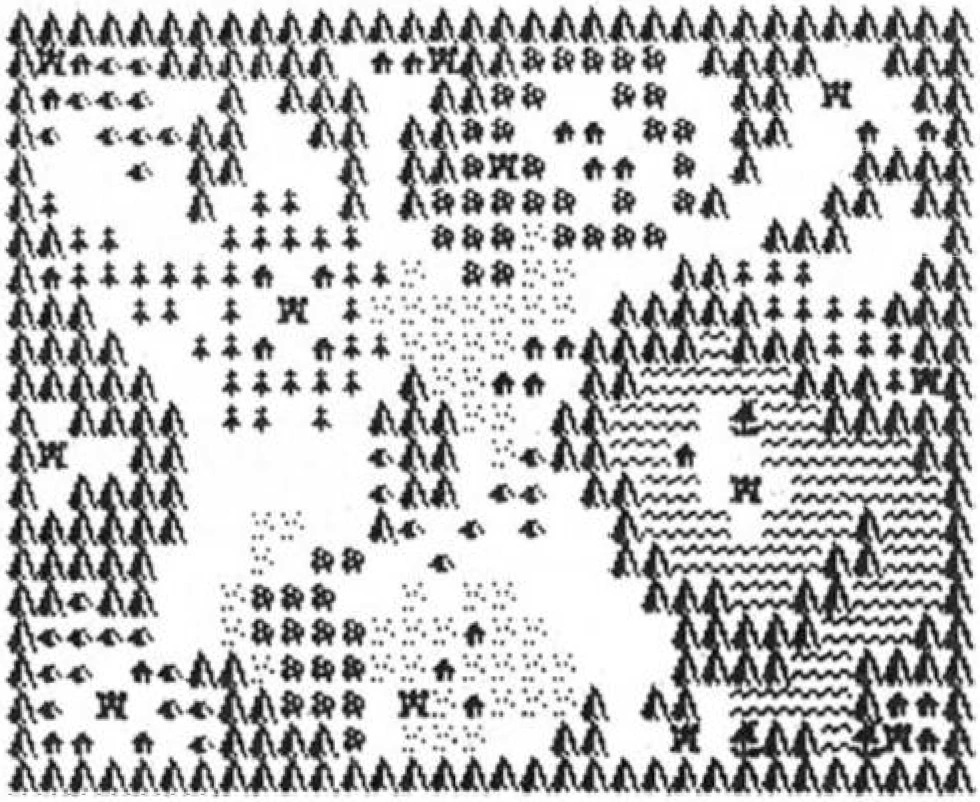


FIGURE 3
Guide to key presses for map editor.

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AFEBBAAAAAA EEFAADDDDD AAAA  AAA
AEBBB  A AAA  AADD  DD  AA F  AA
AB BBBAA  AA AADD EE DD AA  E EA
A   B AA   A AADFD EE D A   AAAA
AH    A HH A ADDDDD D DA   AA AA
AAHH   HHHHH  DDDGDDDD   AAA   A
AEHHHHHHE EHHG DDGG   AAHHH   AA
AAA HH H F HGGGGGGG AAAAAHHHHAAA
AAAA  HHE EHHGGGGEEAAAACAAAHHHAA
AAAAA  HHHHH AGGEE AACCCCCAAAHFA
A AAAA HH H AAAGG AACCC ICCAAAAA
AF  AA      BAA GBAACCE  CCCCCAA
A AAAA      BAA BB AACC F CCCCCA
AAAAAA  GG  AB B B  ACCC CCCACCA
AAAAA   G DD  B     AACCCCCAACCA
AABAA  GDDD  G GG    AAACCAACCCA
ABBBB  GDDDDGGGEGG    AAAAACCCAA
ABB EBAAGDDDGGEGGGG   AAAACCAAAA
AB F BBAADDD FGEGGGA AA CCCCAEEA
AEE E BAAAAD GGG  AA  F IAACIFEA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

square around using keys 5,6,7, and 8. OK?

Fine — now press ENTER, and the cursor will start flashing, ready for you to insert a graphic character. Press a letter key between "a" and "i" inclusive, and the corresponding graphic character will appear at the cursor position on the screen, in the correct colour. Finally, try passing CAPS SHIFT/O to delete it.

Going to war

Now that you're the proud possessor of a map editor, you're in a position to build up the map for "War of the Shires". Figure 2 shows you what the map should look like at the end (except that yours will be coloured, on screen), and Figure 3 shows you which keys you need to pressat each cursor position. the blank spaces, by the way, require no action on your part; just leave them blank. At any stage you can return to the main menu by pressing "q", subsequently saving the map array to tape using option 2. At the start of another session you can load in a partly completed map using option 1, and then carry on where you left off. Go on, admit it — ths is a lot more fun than typing in 704 numbers, isn't it!

To a certain extent you don't need to keep rigidly to Figures 2 and 3 — you can shift forests, hills, meadows and villages around as much as you like, really. However, do please check the positions of the *keeps* very carefully (look for the letter "F" in Figure 3). The keeps *must* be in precisely the locations shown in the Figures, and there must be 12 of them. Mountains will limit the movement of armies in the final game, and so you should make sure that every location at the edges of the map contains a mountain. You'd also be well advised to avoid making changes to the region surrounding the keep of Darkness on the western edge of the map (say within seven or eight locations from the keep) for reasons which will be obvious later on. Also, make sure that at least one ship is accessible from the mainland at the start, or your armies will be left high and dry! Within these limits, there's no reason why you shouldn't build up a small library of map arrays for use with the game.

Keep the machine code bytes, the UDGs, and your completed map array safely on tape — they're all vital. Till next month, when the action starts, happy map-making.....

CROSSFIRE

The elusive 128 Keypad tracked down at last and a chance for you to nominate your choice in our quest to find The Greatest Unfinished Game of all time.

Keypad not required

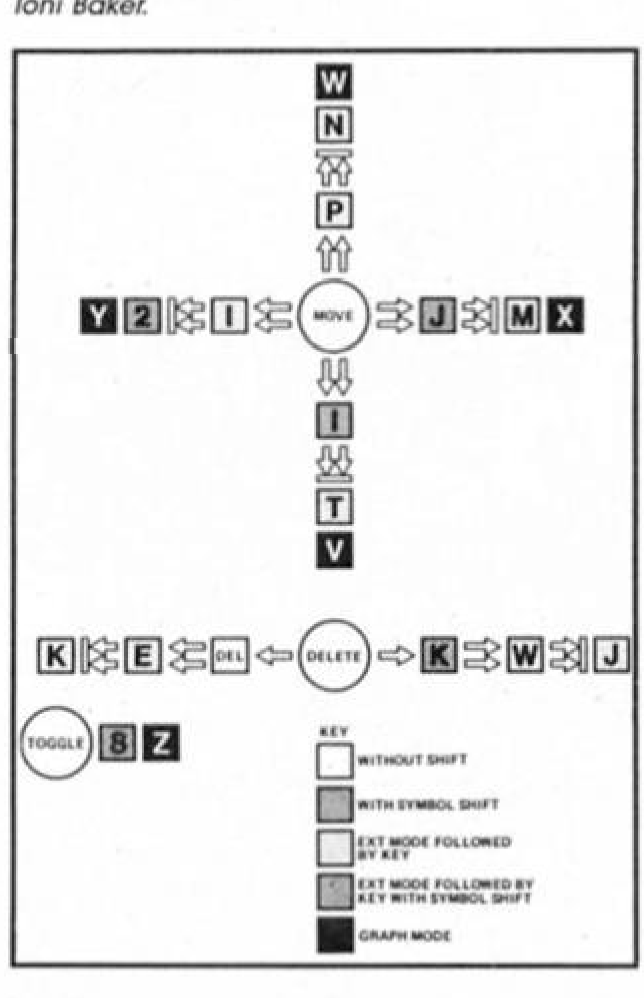
In the June issue of ZX Computing there was a program I wrote for a machine code interrupt routine which put all of the new keys from the not yet available plug-in keypad for the Spectrum 128 in sensible positions on the Spectrum 128 keyboard.

An interrupt routine, however, is not necessary at all! All of the keypad keys are already on the Standard Spectrum 128 keyboard, although in rather peculiar positions. The diagram accompanying this letter shows exactly where these new keys are. All of the symbols from the keypad are visible in the diagram, and by each symbol is a coloured square containing a letter or number (or in one case DELETE). The colours tell you which combination of shift key and extend mode is required with the letter or number in question. For instance, the symbol (which means CURSOR DOWN TO END OF PROGRAM) is accompanied by the letter T in a green square, and the letter V in a black square. A glance at the key will tell you that a green square means EXTEND MODE FOLLOWED BY; therefore, pressing EXTEND MODE followed by T will move the cursor down to the end of the program. Similarly, since a black square means GRAPHIC MODE, it follows that pressing GRAPH followed by V (followed by GRAPH again to exit graphic mode) will have the same effect.

If you wish to make use of the new editing keys, but can't be bothered to load in and set up an interrupt routine program each time you switch the machine on, just cut the diagram out of the magazine (or re-do your own copy) and stick it somewhere near your Spectrum 128 so that you can refer to it whenever your Speccy is in use.

Of course, perfectionists who don't like the idea of having editing keys all over the place with strange combinations of shifts, would probably prefer to use the interrupt program, since that locates all of the new keys in sensible positions (on the cursor keys in conjunction with TRU VID and INV VID) so you don't need a table to remember everthing.

Toni Baker.



Start to finish

When it comes to games software is it a case of many are played but few are finished? In the world of books there are "unread classics", tomes that sit proudly on the owners' shelves but are never looked at. Are there games which have totally bamboozled you and been left halfway through or adventures that have left you totally stumped and languish somewhere at the back of a cupboard?

Here at ZX we are announcing an amnesty on games and adventure playing and readers are invited to point the finger at those games which have frustrated them most. Fill in the coupon below and from your responses we will be able to identify the most unfinished game and adventure of all time. Any additional comments on games playing are welcome and will form the basis of an article in a future ZX. Please send your nominations by December 5th 1986. Senders of the first 10 coupons drawn out of the hat will win a prize.

Great Unfinished Game Quest

My nominations for the Great Unfinished Game are

1...

2...

My nominations for the Great Unfinished Adventure are

1...

2...

What proportion of games and adventures that you buy do you play right the way through to the bitter end?

....

Name...

Address...

....

....

Send to Great Unfinished Game Quest, ZX Computing Monthly, 1 Golden Square, London W1R 3AB.

CROSS WIRES

Ray Elder tackles more readers' technical problems.

This month we really have a variety of obscure enquiries and due to their unusual nature, such as the operating commands of an ancient disk drive interface or the special commands of the American TS computer, we hope that perhaps YOU can help our desperate readers. . .

Dear Sir,

Q *I have the following question to ask. I recently bought a Viscount disk drive interface for my Spectrum at a sale in a local computer shop.*

But they didn't have a manual to go with it and they didn't know how or if it worked. I bought it anyway as it cost me only 10 guilders (about 2 pounds) but now I am trying to find out what commands to use with it.

It seems to work all right, it didn't wreck my Spectrum when I connected it and gave me a message on powering up. The message was "(c) 1983 by Marconics Ltd."

A friend of mine tried to find out what commands to use by reading the contents of the eprom of the interface but couldn't work it out.

So I hope you can supply me with the commands or an address of Marconics Ltd.

P.S. I enjoy your magazine very much.
Yours faithfully
Daniel Muller, Haerlem, The Netherlands.

A I am afraid that the Viscount was just about the least popular of all the Spectrum Disk units, but at two quid you've got a bargain!

We have no address for Marconics they seem to have disappeared, but if any of our readers owned such a beast and would send us the operating details we would be only too pleased to forward them to you.

Dear Sir,

Q *Last month I read an article about a brilliant program sold in America similar to the CARD DESIGNER for the Commodore 64, called "Greeting card designer" for the TS2068 BY ZEBRA systems, Woodhaven, New York, which I sent for and back came the program with a catalogue full of goodies or so I thought.*

I was told the Timex 2068 computer was the same as the SPECTRUM 48K, however after the program loads I come up against two problems. The first is there are some different characters in the listing that my Spectrum doesn't understand (enclosed is an extract of the listing), the second problem is that the program doesn't allow for the use of my interface, only AERCO, TASMAN-B, TASMAN-C and A&J. Is there any way I can convert the program to work with my set-up. I have a SPECTRUM 48K+, "ZX LPRINT III" interface and a "EPSON LX-80" printer.

Your assistance in making this program work for me would be more than appreciated.
Yours sincerely
R.A. Williams, BFPO29.

A I am afraid that there are several differences which make the USA computer and the Spectrum incompatible. Timex were rather unhelpful in supplying a TS machine for us to work with and so we really know very little about it.

BUT, we have many readers in the USA and perhaps one may help us. The Characters which cause the confusion are a curly bracket and a copyright sign which occur after a full colon.

You could try simply deleting them and hope that they are something unimportant such as sound commands, if you rerun your computer then you'll soon find out!

I would suggest that you try this and perhaps add a line 2185 LPRINT CHR$(0);CHR$(4);CHR$(13)

This is the interface initialisation commands and in that line it should set it up ready for the programs use. A straightforward COPY command replacing one of the print routines could solve the problem.

Without my having the software to try out I'm afraid that you'll just have to experiment. . .

Dear Sir,

Q *I wonder whether you can help me. I have a ZX Spectrum+ and an Alpha com 32 printer. All very satisfactory until recently I purchased, after reading an article in ZX Computing by John Wase, a Discovery I disk drive.*

I immediately started having problems and found that these only occurred when the printer was plugged in. I was unable to load commercial tapes even from tape and could not convert Tasword II as advised at the end of the Tasword manual. The command LOAD""CODE would just of be accepted. Without the printer all worked as it should.

I wrote to Opus before I discovered about the printer and they just advised me to buy multiface 1. I have written again since but so far they have not been able to help.

I have also telephoned Dean electronics to ask if they knew of any reason of the apparent incompatibility. They have supplied me with a diagram of the pin connections to the plug but this is not a lot of help.
Thanking you in anticipation,
Yours faithfully,
R.J. Lawman, Cardiff.

A This has been a problem which arose after Alphacom had been supplying their printer for some time. It is simply not compatible with some peripherals and unfortunately the Discovery is one of them!

The only real advice I can give you is to buy a Centronics type full spec printer such as the GLP 2 or the Shinwa CP80.

I know it will set you back around £100-£200, but if you intend using your computer for serious text output them you will never regret it.

Dear Sir,

Q *I own a ZX Spectrum 128K. When using it (programming or game playing) I get flickers of interference across my television screen.*

The flickers last only an instant and come every seven seconds. They appear as silvery lines and in no continual place. As they are very distracting could you please tell me if this is normal and if not is it my computer or my television that is at fault.
Yours Sincerely,
Richard Finch, Southampton.

A Probably neither.

This sounds like a case of interference from some other piece of electrical equipment. It could be that the computer is too close to your TV or that the PSU and the computer or the TV are interfering with each other.

Try moving each unit to different positions, but in the end I strongly suspect that another TV or a Cooker, Video, Fridge etc, is causing the problem. The only sure way is to turn everything off and just have your computer and TV on. Then again, a neighbour's unit might be the source of the problem.

Often one of the special suppressing plugs will cure or help allieviate the problem. It's a case of trial and error yet again!

A very helpful letter was received on the topic of the Opus and Printers from Mr. Turner of Hull, it reads:

Dear Sir,

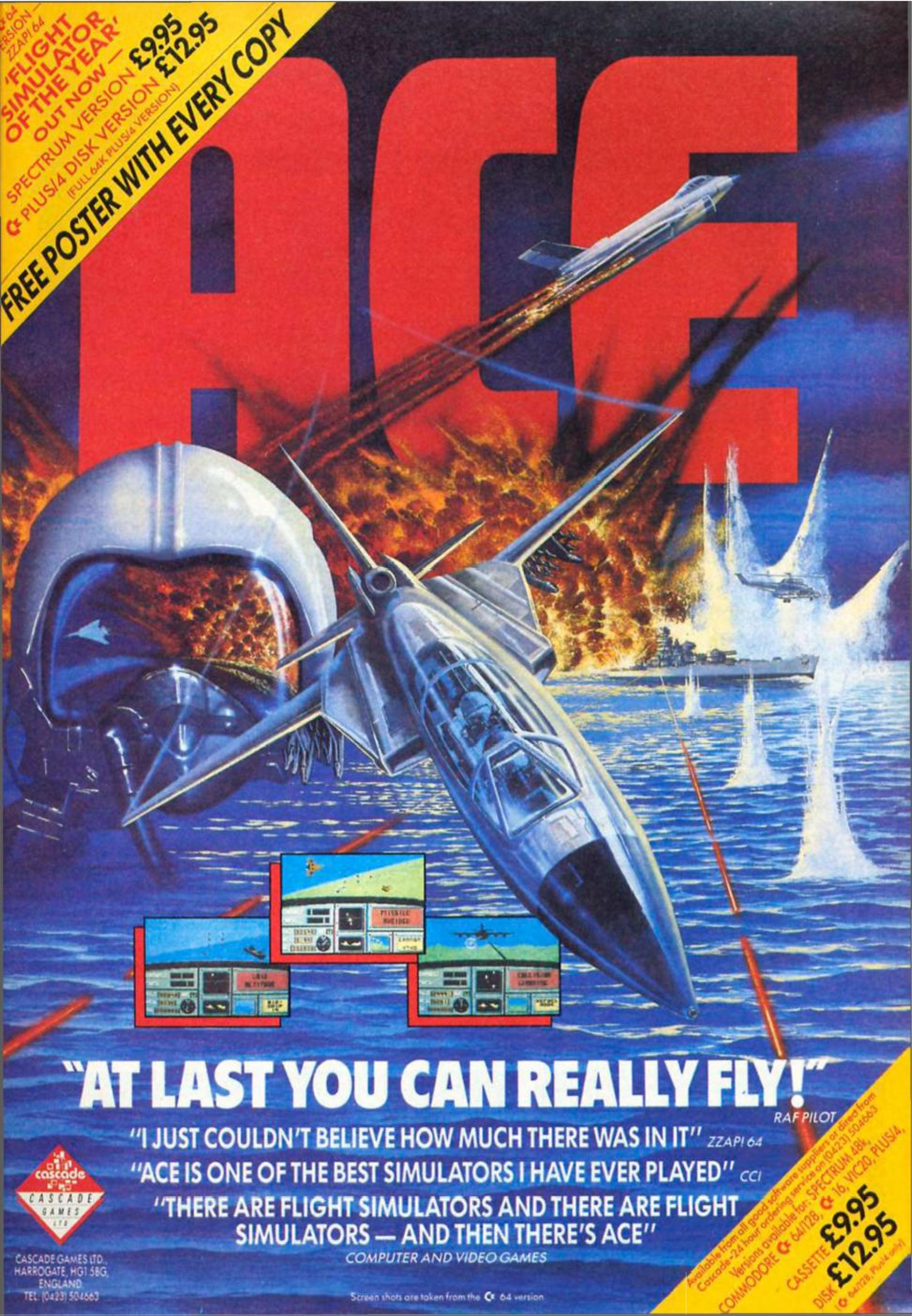
A *Reference Janet Watson's query in August edition of ZX Computing. I too have the Opus Discovery and Brother M1009 printer and also experienced problems using Tasword 2. After hours of work I finally contacted Tasman Software Ltd. They were most helpful.*

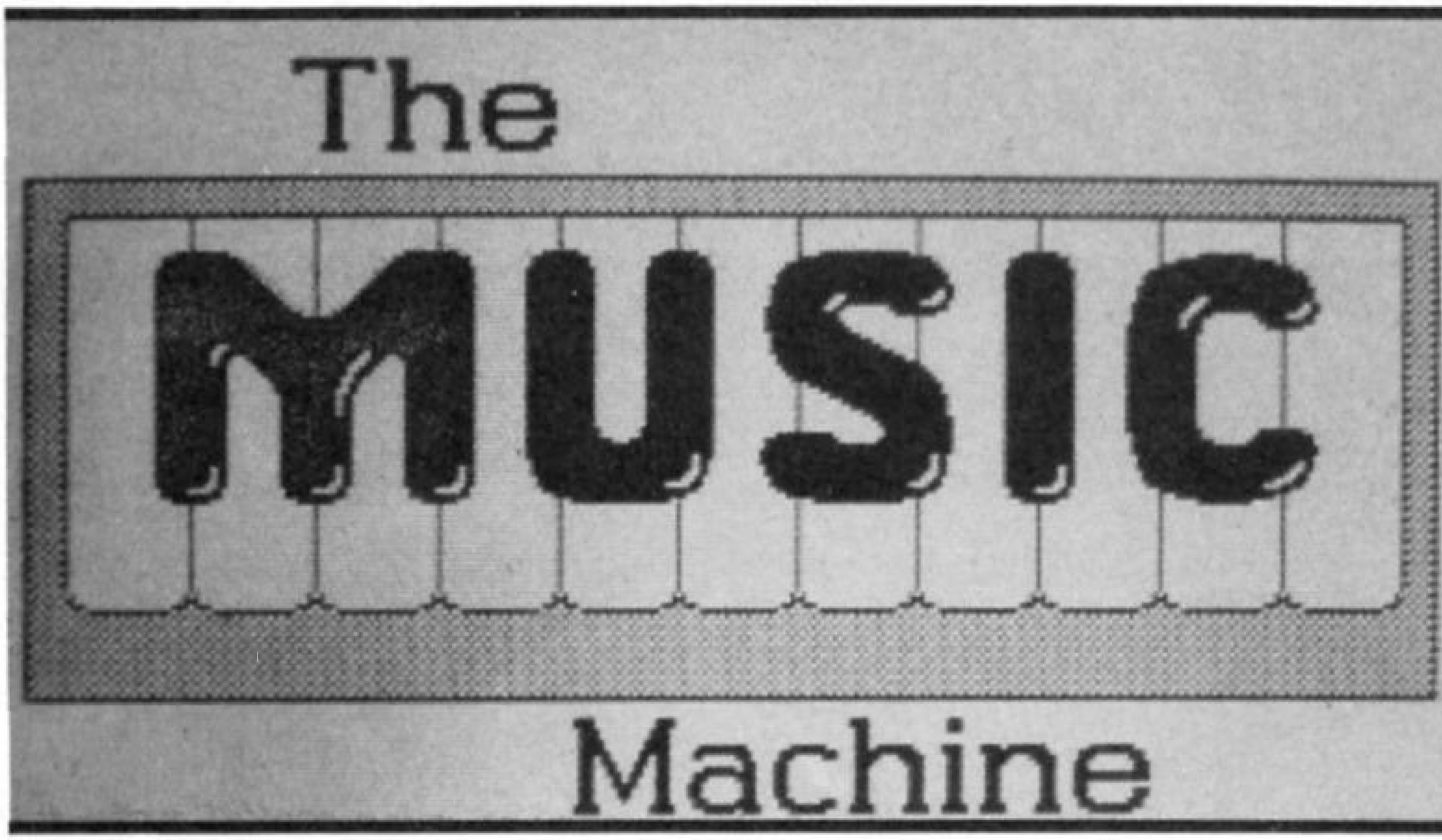
The answer is Discovery will accept the microdrive version provided the following alterations are made to the basic program.

ADD LINES
6 CLOSE # 3
8 OPEN # 3,"B"
DELETE LINES 279 AND 281.

This is because the Discovery will not accept FORMAT "b";X Where X is the baud rate.

Printing may be stopped as normal by pressing and holding 'q' but if BREAK is used then you must restart by executing GO TO 6 or you will lose the text. You will also have to start to print from the beginning again.
Yours Faithfully,
T.W. Turner


ZZAP! 64
'FLIGHT SIMULATOR OF THE YEAR'
OUT NOW —
SPECTRUM VERSION £9.95
C= PLUS/4 DISK VERSION £12.95
(FULL 64K PLUS/4 VERSION)
FREE POSTER WITH EVERY COPY
ACE
"AT LAST YOU CAN REALLY FLY!"
RAF PILOT
"I JUST COULDN'T BELIEVE HOW MUCH THERE WAS IN IT" ZZAP! 64
"ACE IS ONE OF THE BEST SIMULATORS I HAVE EVER PLAYED" CCI
"THERE ARE FLIGHT SIMULATORS AND THERE ARE FLIGHT SIMULATORS — AND THEN THERE'S ACE"
COMPUTER AND VIDEO GAMES
cascade
CASCADE GAMES LTD
CASCADE GAMES LTD., HARROGATE, HG1 5BG, ENGLAND. TEL: (0423) 504663
Screen shots are taken from the C= 64 version
Available from all good software suppliers or direct from Cascade–24 hour ordering service on (0423) 504663
Versions available for: SPECTRUM 48k, COMMODORE C= 64/128, C= 16, VIC20, PLUS/4,
CASSETTE £9.95
DISK £12.95
(C= 64/128, Plus/4 only)



Ray Elder enthuses about the potential of Ram's new comprehensive musical add-on.

The Music Machine
Ram Electronics
£49.95

This smallish uninteresting looking black plastic box is perhaps the most powerful, versatile and exciting peripheral that I have had the pleasure to review for ages!

Described as 'The Complete Home Computer Music System' I consider this claim to be a modest appraisal of the unit, it could be used in professional applications as well. So what does it do?

At Home

The great majority of purchasers will go for its superb sampled drum sounds and built in sequencer allowing complex and carefully constructed patterns to be created bar by bar and linked together to form a complete song.

A very versatile unit the quality and flexibility of which is equal to the £250+ Yamaha dedicated RX21 that I use. On the plus side is the fact that each bar can be set to individual tempos, not feasible with the RX. But there is a real time play mode where your fingers can attempt to mimic Buddy Rich. On the minus side is there is no "real time" pattern constructing and only two Toms, the RX has three. However the Music Machine has Cowbell and the RX hasn't.

But this does not matter!

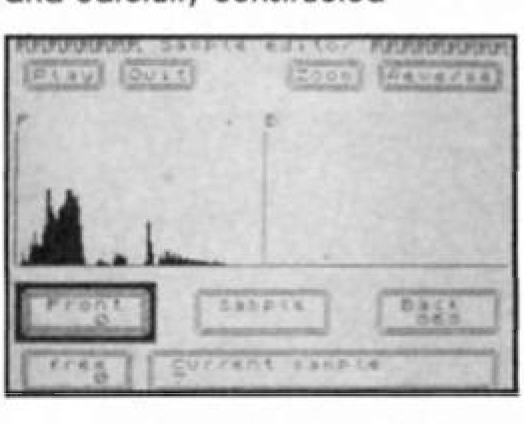
If you want another Tom then you can have it, or remove the cowbell or any of the sounds and replace them with any other you fancy because the Music Machine is also a Sampler! This means that you can record digitally any sound you like via the cheap microphone supplied, tape or line out of an amplifier.

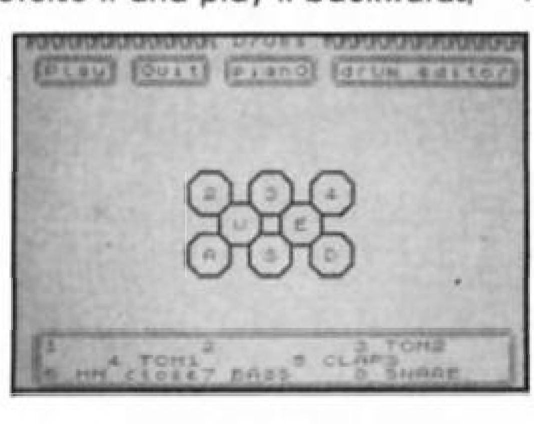
Once a sound is in memory then you can set the start and end parameters and play as much or as little of it as you wish, even looping it for continuous sustain. You can reverse it and play it backwards, interesting, or go to the "piano" screen and play it back over a 12 note, one octave range rising from middle C.

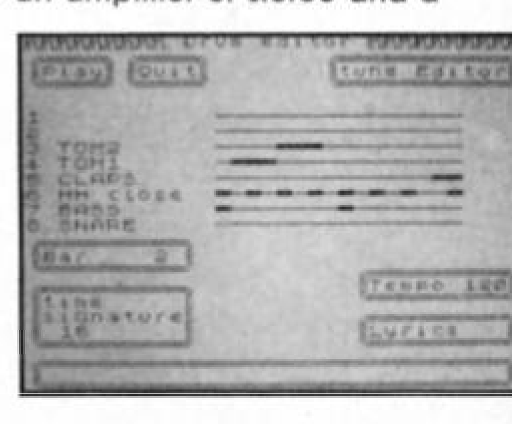
Once you have exhausted the novelty of that then you can use the tune sequencer and play it over a much extended three octave range and in two parts. It is here that some of the limitations of an inexpensive unit may become audible, the sustain effect is played by a very fast staccato repeat and it can be heard as such, also the tone may become unpleasant in the extremes of the range. A £10,000+ Fairlight allows multi samples across the whole range to be taken so what can you ask of a unit such as this?

For technicrats the sampling rate is 19.444KHz and this gives approx 1.1 secs, it sounds short, but in fact it gives plenty of time to say "Samantha Fox" (should you so wish). A start and end of a sustain loop within the sample feature would have been useful, and the enlarged display of the waveform often resembles a burst from an airbrush and this is a pity.

The rear of the unit bristles with sockets and the three which most owners will use are the microphone IN, the Phono OUT to an amplifier or stereo and a









Headphone for personal listening. Only the output from either the tune sequencer or the drum sequencer can be sent to the headphone or phono socket at any one time.

Not just simply a fascinating toy as are most of the samplers I have seen, including some made by well respected companies in this field, but actually usable to create genuine musical compositions.

For anyone with musical interests this is an absolutley essential piece of equipment, I've heard 'Rap' records with less backing than this can produce, and the quality is good enough for studio use.

Midi

Should you be one of the growing number of serious home musicians who own a keyboard such as the Casio CZ101 or any of the other instruments fitted with MIDI (Musical Instrument Digital Interface) then you can greatly extend both units' use.

There are three MIDI sockets fitted, standard five pin DIN, for MIDI IN, OUT and THROUGH and via these you can either play your sampled sounds over the full keyboard range, sync with external sequencers, play the Music Machine's sequencer out to the keyboard (at the same time as using the drums via the headphone or phono out) or any other combination you desire. This is possible from the software which operates in both the common Omni and Poly modes allowing full channel assignment and internal or external clock control.

Easy use

All this in one unit and program must make it complicated to run.

Not a bit! The manual is written to suit all abilities, step by step chatty approach throughout but with full technical detail for those with deeper understanding — something many of the dedicated "professional" units I've looked at do not usually give.

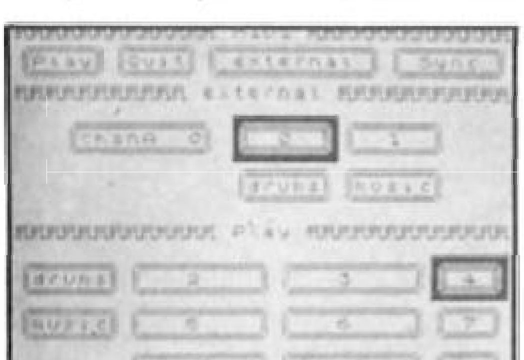
As for operating, the menu system has been carefuly designed with many options being consistent whether you are in the Sampler, Piano, Midi, Echo or any of the many other operating screens. You soon learn the essential commands and the others are nearly all self evident.

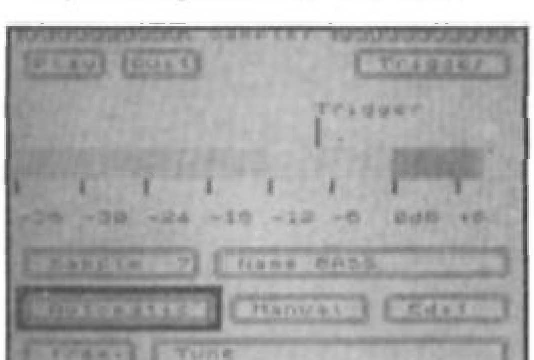
I am afraid this review is rather on the enthusiastic side, but then I have deliberately compared this unit with others of much higher price and it holds its own. A simple MIDI interface by itself can cost over £100, this has much, much more to offer.

RAM ask for ideas, there is no limit when you consider MIDI, for example how about; A real time multitrack sequencer with auto correct and variable quantize, or a multitrack step time sequencer, both with high resolution printout facilities, or a midi patchbay system (give me a unit and I'll write one of those!) or, or .... the list is vast indeed.

Yes, you may say, but it'll cost the earth and be well out of my reach. I agree it is a little more than the average £15-£30 interface but at an astounding £49.95 it offers incredible value for money, the chance for anyone with an interest in music to get into the latest hi-tec musical development and I have no hesitation in giving it the highest accolade possible from a hard bitten, cynical reviewer.

I will buy one.







joysticks

Joystick connoisseur Charlton Appleby takes a critical look at established Spectrum joysticks and also playtests Cheetah's new 128+2 compatible sticks.

125+ Cheetah £8.95

The Cheetah 125+ joystick is a pistol-grip type of device with a thumb button on the top of the stick and a trigger for your fore-finger. There is also the addition of two buttons on the ample, rounded, base. The moulding of the stick is very curvacious and fits the hand nicely, the base has four suction cups on it to secure the stick firmly to a smooth surface.

The cable is about a yard long with two plugs on the end. This split adaptor also features on the Mach 1+ and is of special interest to Sinclair owners, especially prospective purchasers of the ZX Spectrum 128+2. One of the plugs is a normal Atari-type D9 suitable for the majority of home computers, and the other one is a grey connector for the side of the new Sinclair machine. This second connector is re-wired in the manner that the new machine needs to get its signals, so you don't have to buy those awful SJSI things. My only worry is that, whilst this is a good idea, the way that the first connector has two wires coming out of it may weaken the link at the plug attachment point. As this is an area where a lot of joysticks break, and short of cutting the cable and replacing the plug, it is irreparable.

Also on the stick is the now almost obligatory, auto-fire switch. This is a switch to make the joystick function like a machine-gun, and to save wear and tear on the fire buttons (an area where a joystick is likely to fail first). A lot of modern games have auto-fire detectors that will cut out this facility or make it function to the detrement of the player, but a lot of games wll be made more playable — in fact, some games are almost impossible to contend with without an auto-fire feature.

My main gripe with this stick is that it is not very responsive, the internal construction is that of collapsible domes, which usually give the best feedback, however in this case the stick is almost dead in my hand and games can be very tricky to play.

However, I would definitely recommend the 125+ for owners of the Spectrum 128+. It has all the features you would expect from an up to date joystick and although a bit lacking in the response department it's an efficient multi-purpose stick and at £8.95 it's a bargain.

Mach 1+



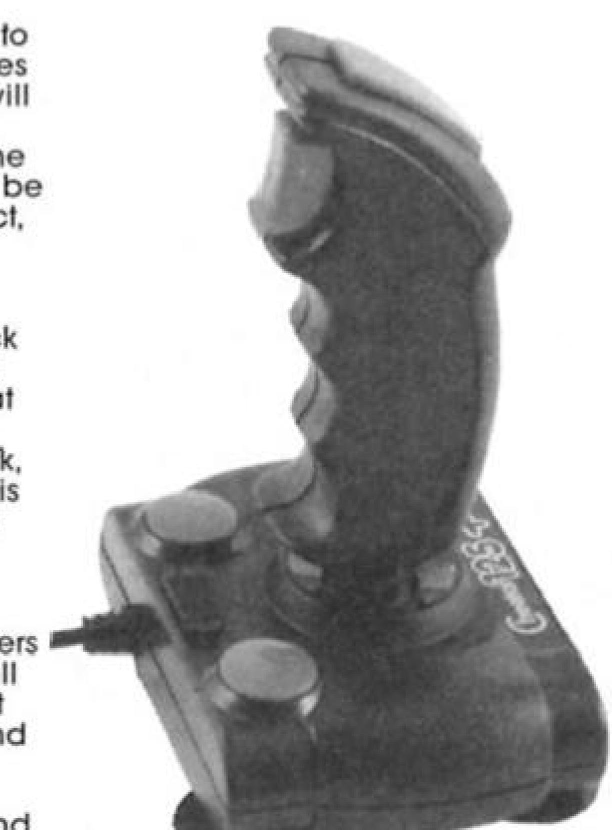
Mach 1+ Cheetah £14.95

The Cheetah Mach 1+ is similar to the 125+ in that it has four fire buttons: two on the base and two on the stick, and in the same positions. However in this case the body of the stick is much bigger and angular and thus more difficult to use and a lot less comfortable than is expected for this kind of money.

Otherwise, the features of the other model are retained — two plugs, auto-fire, suction cups, etc. The internal construction of this stick however, is with micro-switches. These will last virtually forever under normal conditions.

The stick is awkward to hold and so is the base. Micro-switches are not very responsive at the best of times, and that point is proved again with the Mach 1+.

Apart from being very reliable it has little to recommend it over the 125+. If you want a joystick that will never let you down then you've found it with the Mach 1 but for enjoyable play opt for the 125+. The substantial price differential (£6) may also be a big influence on your choice.



Ever since 1976 when the first Atari VCS reared its ugly head in a few pioneering homes, the joystick has been a major bone of contention. Within a couple of years, third party companies were producing alternatives to the Atari standard stick. When the computer took over the Atari D9 standard became the *de facto* method of connecting a joystick or tracker ball to a computer. This standard persists to his day with even the mighty machines like the Amiga and Atari ST having the same little socket on their sides.

All Spectrum joystick interfaces have this D9 connector in them, some even have two. The one notable exception is the new 'Amstradified' Spectrum 128+2, which has had the wires all jumbled up in an attempt to make you buy Amstrad/Sinclair Joysticks. For the purposes of this article I will assume that anybody with one of these machines has either got an adaptor, or forgotten that they exist at all and has gotten himself a Kempston interface or something similar.

So what do you shove into that hole in the side of your machine? Having been playing games with joysticks for the last eight years or so I can safely say that I have some favourites. You may not agree with me, but then joysticks are like golf clubs, different sticks perform a different job, so select one that matches your needs.

Atari Standard
Atari
£6.95

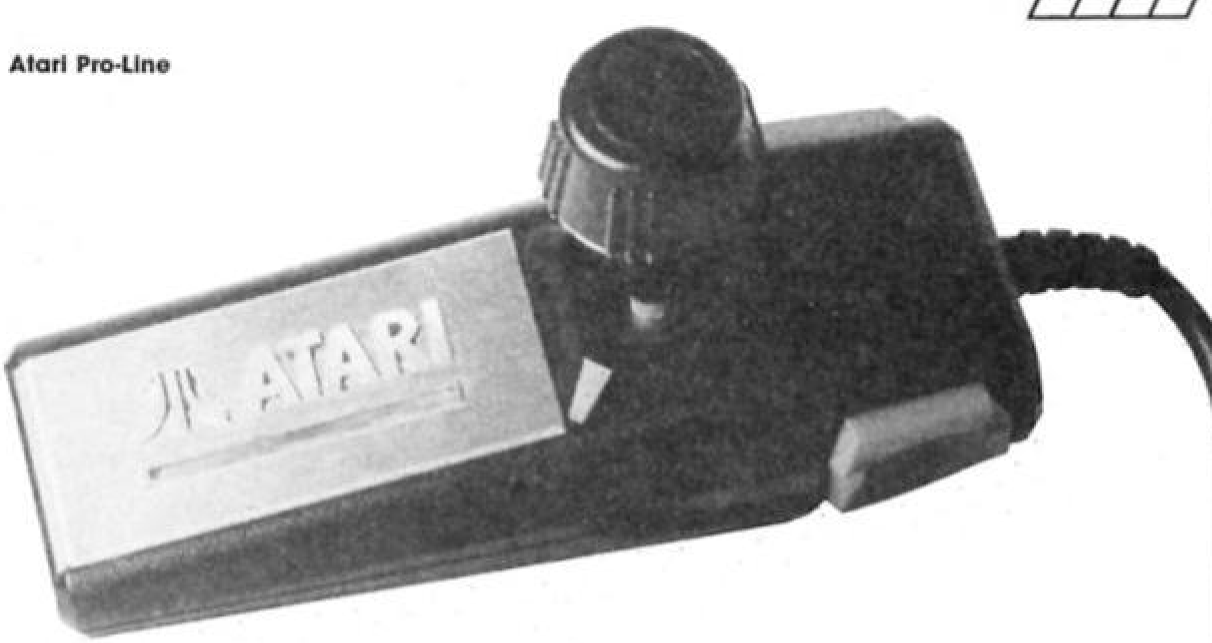
The original, and some say the best. This awkward looking and ungainly stick is notorious for having started the whole ball game off. Internally it uses collapsible dome technology to make the connection, deforming a dome of thin metal until the electrical switch is made.

This system has the advantage of high tactile feedback, but the disadvantage of poor reliability and a short life time. These sticks seem to need a 'bedding in' period before they work to their best efficiency. The greatest advantage is that they're cheap, however their shape is very angular and the corners can dig into your palm and hurt a lot after a marathon all-night session.

Atari Pro-line
Atari
£9.95

The successor to the previous stick, this controller uses a contoured grip with two fire buttons for left/right handed operation, or fore-finger/thumb interchange. The short stick is

Atari Pro-Line



steel and unlikely to wear out, and is incredibly responsive.

The contacts are still collapsible domes, but this seems to combine with the short shaft to make this one of the most responsive joysticks that I have ever come across. This is my all-time favourite general purpose stick, capable of playing any game that there is, was, or could be!

Quickshot II
Spectravideo
£11.95

Probably the best known stick in the business. The Quickshot II has a contoured grip that fits the hand in a manner so perfect that it is a miracle of design. Two fire buttons are provided, one in a pistol trigger position and one on the top that fits your thumb. The base has four suction cups on it to grip the stick to a flat, smooth surface for one handed operation. Also on the base is a little switch for 'autofire' operation. This switch connects to a small circuit inside the joystick that creates a machine-gun effect when using the fire button. This is achieved by pulsing the fire button's output to the computer. This allows multiple firing in a game that allows such things, though a few of the more modern games can detect when an auto-fire switch is being used and will disable the fire button all together — with disastrous results.

Atari Standard



Internal construction is with collapsible domes, though with the large amount of leverage that such a stick affords, you may find that in some circumstances this stick will not have a long-life span — especially with younger players.

Fighter Pilot
Tandy
£8.95

This stick used to be marketed by Spectravideo under the name 'Quickshot' before the Quickshot II came out. Now re-badged, Tandy are selling it under their own brand name.

This stick has a fire button on the base, one on the top of the stick for your thumb, and a contoured grip — though not up to the standard of the Quickshot II. The base has suction cups on it to provide a firm footing for one handed operation, though there is no great problem using two hands as the base is also designed to be held.

Internal construction is once again by the collapsible dome method, with the usual limitations of not being too hard wearing, but with the advantages of good control and tactile feedback.

Trackball
Atari
£22

Not really a joystick, but can be used as one. The trackball was first introduced into the coin-op arcades on a game called Missile Command, and more recently on a game called Marble Madness.

The track ball consists of a 'billiard ball' mounted in a casing with detectors to report its movement to the computer and a couple of fire buttons. Instead of bending a stick in the required direction ou roll the centre ball. This can give you an excellent control advantage in some games, but can be a liability in others. A track ball is at its best in Missile Command and Marble Madness type games, and can come into its

own in a lot of maze games.
A useful controller to have handy though definitely as a spare, rather than as your only control system.

Speed King
Konyx
£12.99

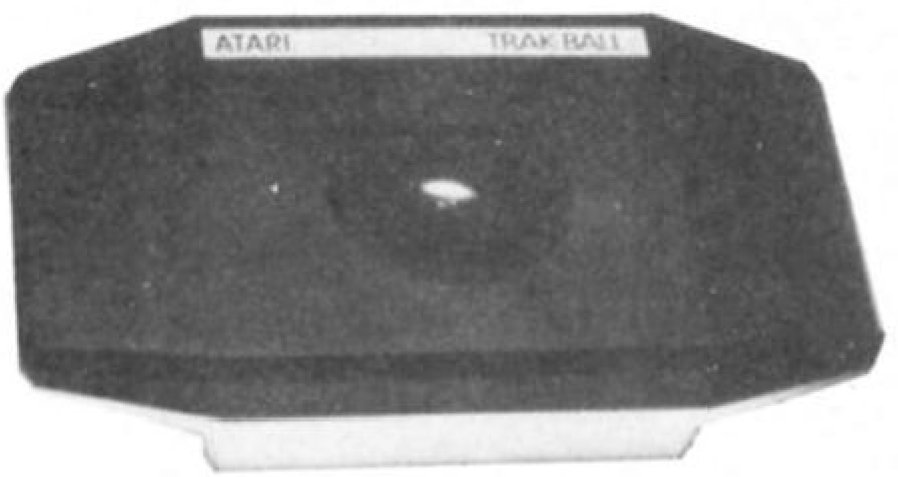
This is a fairly new 'stick' with a less than conventional configuration. An 'ergonomically' styled grip fits into your left hand like it was part of it, with the triger resting neatly beneath the crook of your index finger. The top of the stick has a short, stiff stick protruding from it.

The switching is via micro-switches that will virually last forever. However, their response time is a bit slow, and the tactile feedback is close to nothing. You hear your stick move before you can feel it or see it.

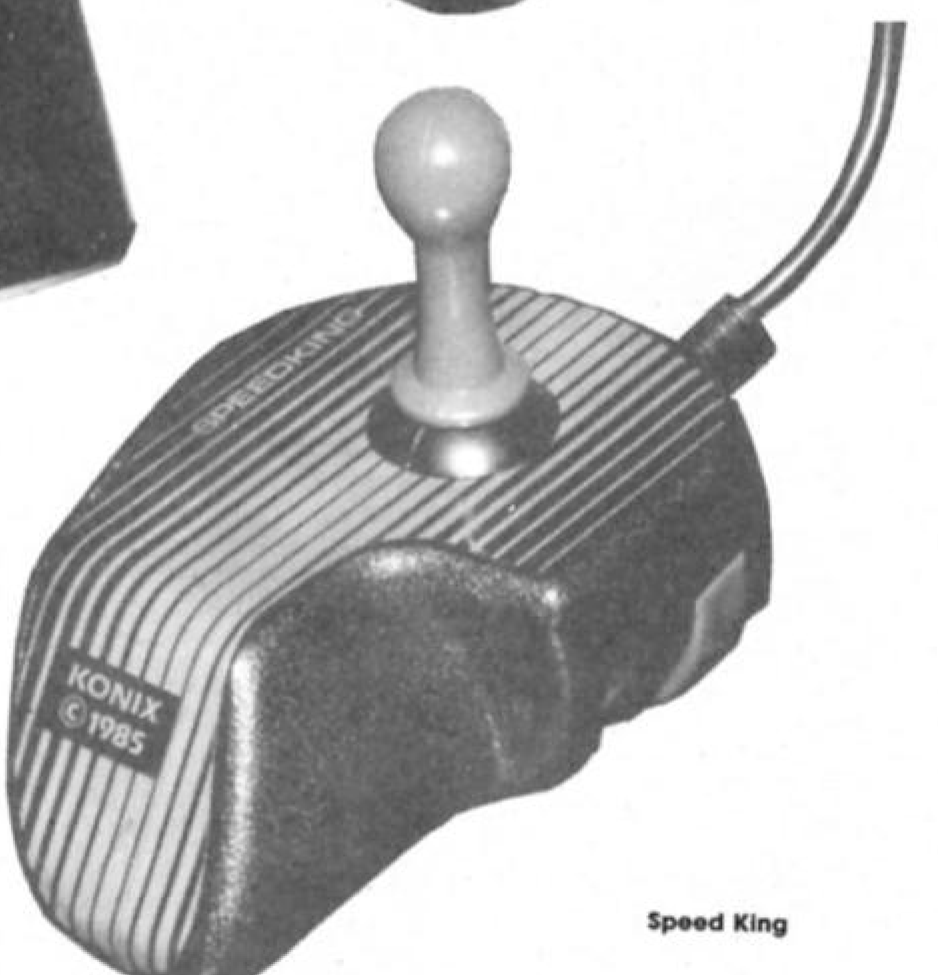
The grip is also a bit dicky at times. When you play for a long time in a difficult game, your hands can get a bit sweaty. With this stick you start to let it slip after a while and the whole situation gets out of control. If you are left handed — forget it!



Quickshot II



Trak-ball



Speed King

Mazemaster
Tandy
£11.95

An interesting concept here, a joystick that can turn off its diagonal movement. Most joysticks allow for diagonal movement, but this stick lets you switch between eight-way and four-way operation. Why? For maze games of course. Ever had that time where you skid sideways into the spider? Well now this joystick will alleviate this to some degree.

The joystick is an awkward square shape, with a little stick protruding above it, this arrangement is fine for mazes, though I wouldn't want to play an action game with it.

Construction is via micro-switch, and should last (effectively) forever, barring any adverse conditions.

Well, these are the sticks that I like to use. You may prefer others, but these have stuck with me through thick and thin and I wouldn't give them up for the world.

HARDWARE

sound sampler

Cheetah follow up the success of their Specdrum with a new Sound Sampler.

SOUND SAMPLER
Cheetah Marketing
£44.95

The Cheetah sampling system consists of an interface, software cassette and microphone. The interface connects to the port at the back of the Spectrum (48K, Plus or 128) and has two control dials on the front, a mini jack sound input socket on the left hand side and a six foot lead fitted with a phono plug to take the output to an amplifier or stereo system. There is a problem in that the lead is not really long enough and the phono plug is only suitable for stereo or expensive mixing desks. The more usual plug is a standard jack, and it might have been better if the control knobs had been positioned on the top and a second socket had been fitted for output.



Sound sampling has become the "in" thing in the music field at the moment due to recent developments in technology and, since it is a computer based technique, it is not surprising that such a device should be developed for our home micros.

To put it rather simply, sound sampling involves making a digital recording of a voice, instrument, piece of music, or some other sound, processing it as required and replaying it in a choice of pitches.

The microphone is a fairly basic one for use with a portable cassette recorder and is not really suitable for 'serious' use.

Although a Microdrive backup routine is supplied, there is a dire warning not to use the interface with anything other than another Cheetah Unit.

Tech specs

The specifications are impressive, replay over two octaves, superb processing allowing very fine selection of start and end points to be made, and sustain start and end points. These are chosen on a graphic waveform display. Samples can be reversed or conbined, saved or loaded and reconfigured at will.

The frequency or bandwidth is 17.5KHz and this is as good as you'll get on some dedicated samplers for many times the money. Sample rate is 70000 times a second and maximum length is around one second. Which is long enough to say your name or a simple Nineteen!

The utility or effects program is great for playing around with, providing echo, reverb, fuzz, chop, bubble, and pitch effects, very noisy though and there's not enough control for serious usage.

Usability

The publicity claims "Home or Professional" use and I wouldn't disagree. All I would say is that the professional use may be rather limited, especially if it was intended for recording as the sound quality is good rather than excellent, and the background noise is rather too high for professional standards even under optimum conditions.

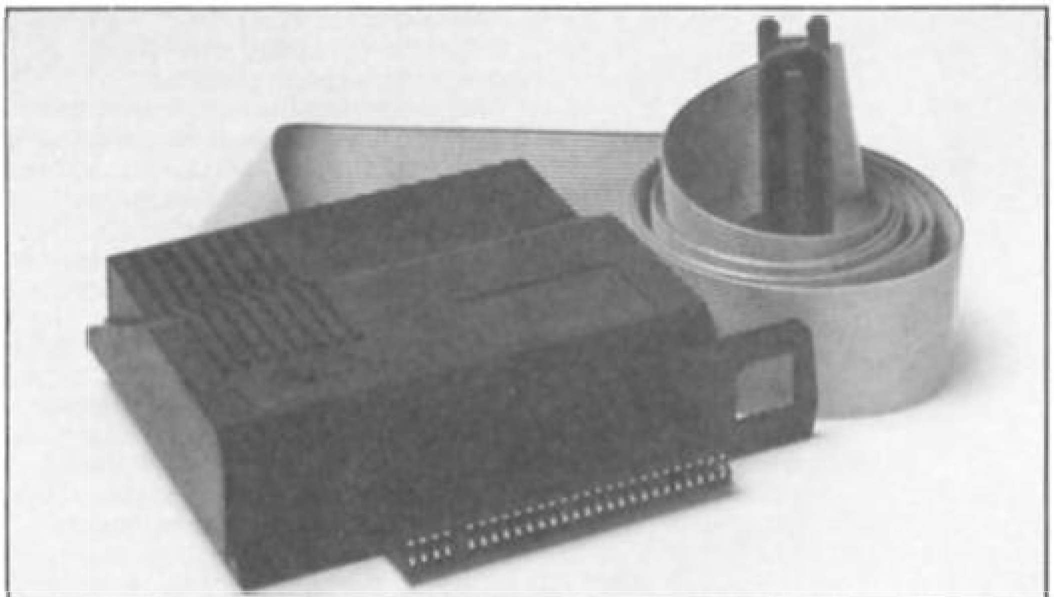
The input section is well designed, allowing a wide range of impedences from microphone, line or instrument sources and the output match proved compatible with both my power amp and my cassette line in.

Operating the sampler is a joy. Everything is menu driven and error trpiing seems to be very comprehensive, even with my idiot act in full swing it informed me of where and why I had gone wrong each time.

I loved it, even at the level of simply playing around it was great. I took it into a local school and the pupils had a great time with the utility program and the Science teacher was devising ways of using it for the new GCSE exams!

Finally, I have spent quite some time using the sampler, and I am becoming more and more impressed with it. I would even cautiously suggest that a studio might be able to use it for some short and limited effects, provided enough noise reduction was available.

RAMPRINT



Not only does RAM's new interface have everything but the kitchen sink crammed into it, the instructions are easy to understand too!

Ramprint
RAM Electronics
£34.95

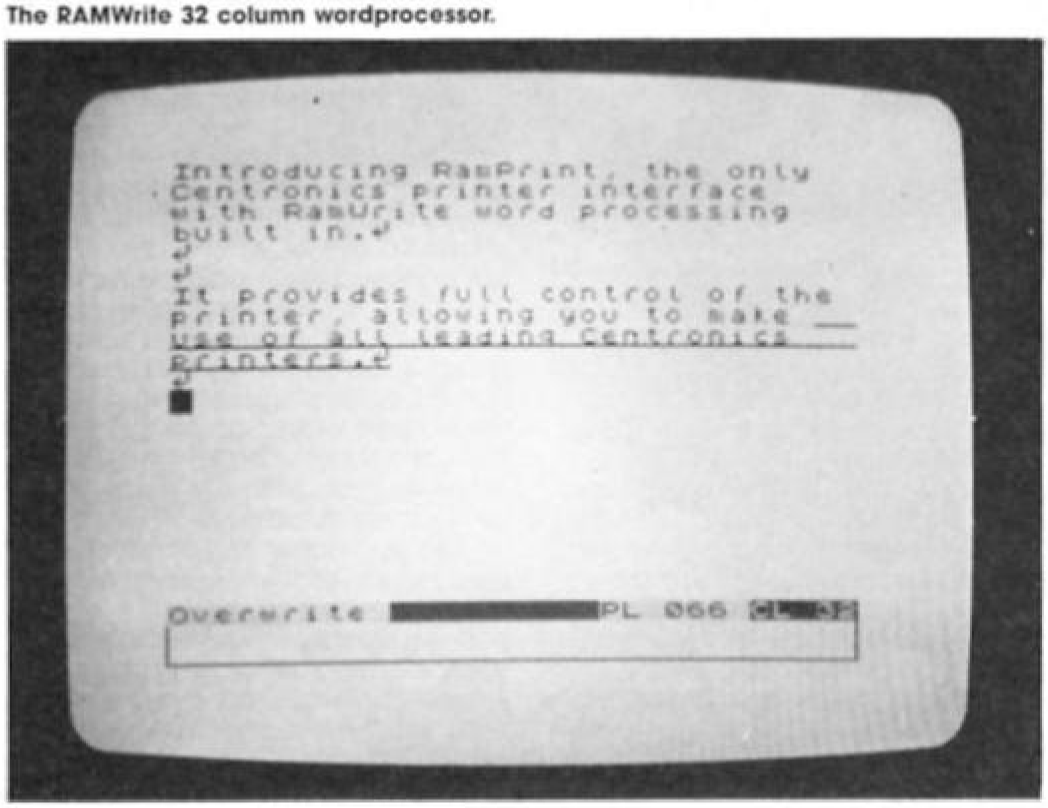
Now let's be sensible about this; what are the reasons for buying a 'real' printer to use with your Spectrum. The main ones have to be to get decent listings, screen dumps and wordprocessing facilities. I got hooked on wordprocessing about a year ago when I discovered Tasword 2, and I would much prefer wordpro to using a typewriter if it weren't for my complete inability to cope with printers. The words always look fine on the TV screen, but I can never get them to come out of the printer at the other end.

The trouble is that you can't just plug a full size printer into the Spectrum and get it to work straight away (you could do this with Sinclair's new ZX Printer but that was very limited in both quality and options it offered). First of all you need an interface and the lead to connect the two machines, then there's the awful business of initialising the interface and shoving in all the control codes. Then, if you're using some sort of wordpro software, you'll probably have to fiddle with that to get it to work with the particular interface that you're using. And if you get any of that wrong the whole setup goes mad and starts vomiting up reams of paper covered in a long-dead Greek dialect when all you wanted was your laundry list.

The whole business is a pain, and it's driven me back to my steam-driven typewriter many a time.

With that in mind it's strange that nobody has yet come up with the idea of producing an all-in-one printer interface with its own wordprocessing software so that the two are compatible and don't require hours of fiddling to get them working together. But that, at last, is what RAM Elecronics have done with their new Centronics-type RAM Print interface and RAM Write software which is stored on a ROM chip within the interface itself, not to mention some of the other features included.

The RAMWrite 32 column wordprocessor.



Kitchen sink syndrome

To start off with, the interface comes with its own Centronics cable built in, so there's no additional expense involved in buying one or working out what to do with it.

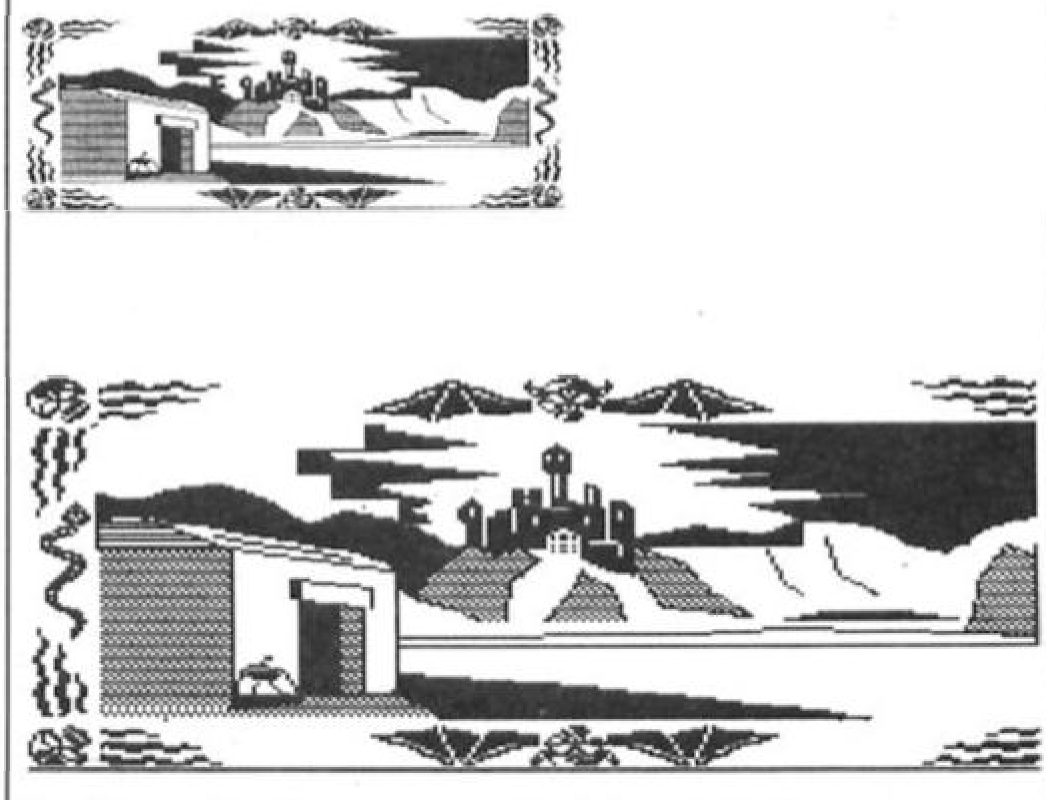
Then, surprisingly, RAM have made the interface emulate the old ZX Printer so that you can use the Spectrum's own LLIST, LPRINT, and COPY commands for listings and screen dumps. this is very sensible as it combines the simplicity of use of the ZX Printer with the high quality printing available from the whole range of Centronics printers.

RAM have also avoided a potential problem here as the Spectrum 128 isn't compatible with the ZX Printer when operating in 128 mode. However if you enter the command PRINT IN 251 on power-up then you can sidestep that particular problem. the commands for the 128 are slightly different, LPRINT "© copy" rather than simply COPY, but that's not too much of a problem.

Now we come to what is normally the tricky bit, the control codes. Thankfully though, RAM, have given you some help here also. Entering the command LPRINT "© set" calls up an options menu, a pull-down window which overlays, but doesn't wipe out, what is already on the screen. The options available here aren't exhaustive but they do at least

cover the most often used options; Line Feed, Tokens, Copy Text, Enlarge and Width. For each of these you can simply choose the On or Off state as required without any of the business of having to look up control codes and enter them as endless strings of CHR$. The RAM Write software has its own set of control codes for different types of printing, but if you want to do fancy things with listings then you'll probably have to go back to looking up the codes in your printer manual. C'est la vie.

As a final touch the interface also includes a Kempston joystick interface — RAM are certainly making sure they've covered all their bases. The interface is well constructed in a robust black plastic. There are no exposed parts and there's even a small ring that goes around the power socket to make sure that everything fits nice and tidily.



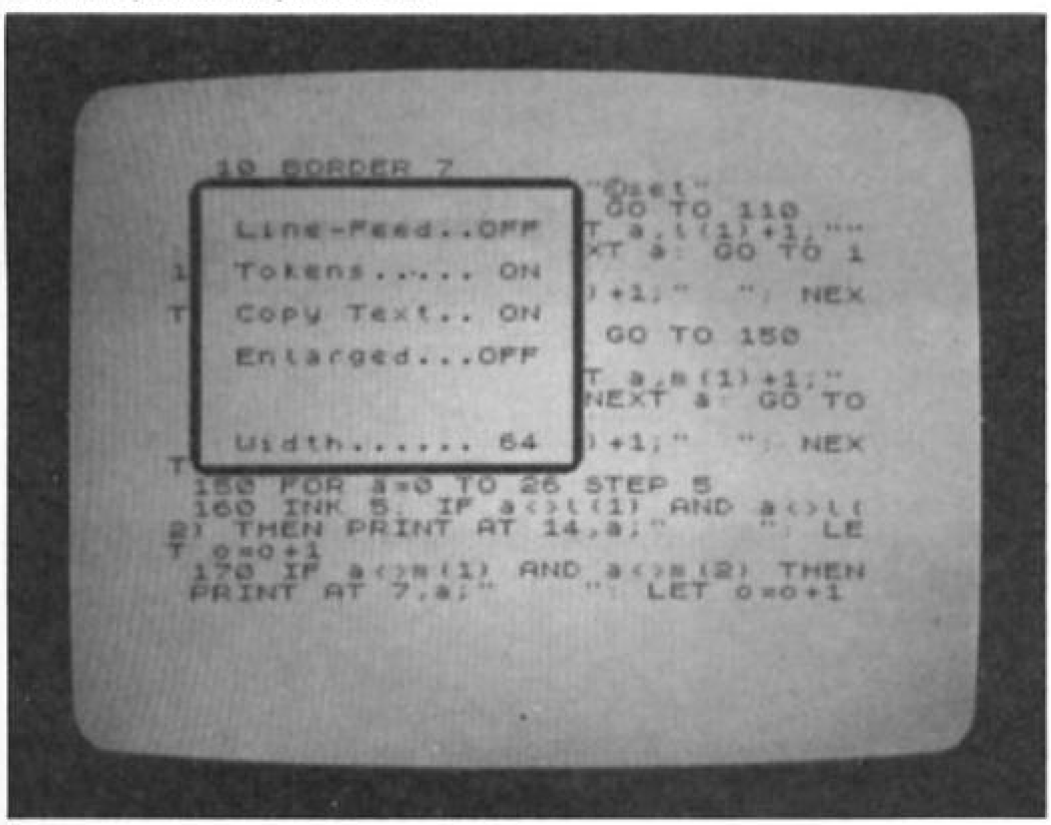
The Enlarge option allows you to copy two sizes of screen dumps.

RAM Writes

Of course, the most obvious attraction of this interface (for me anyway) is the inclusion of a wordprocessor program on ROM and the convenience of instant access and printer/interface compatibility. Entering the command LPRINT "© word" instantly calls up the RAM Write program and you are presented with a screen in which the upper 22 lines are blank (for text entry), while the lower two lines are for the status display and choice of option. When the cursor is on the bottom line you can choose whatever options you want simply by typing the appropriate letter (b= Back to Basic, c= Copy text, d= Display text and so on). The options available cover all the usual string search/replace, block copy/delete, and so on. These aren't as exhaustive as the options available in Tasword or The Writer, but they're adequate for most simple wordpro, apart from the omission of a wordcount (one day someone will produce a wordpro that has its own word count . . .).

Entering 'e' puts you into Edit mode and reveals the only real disappointment in RAM Write. You can only enter text in the Spectrum's normal 32 characters per line, and while the printed output does have wordwrap, 64 character lines, justification on and off, you can't see these things on screen while you're editing the text. There is a display mode which allows you to see the text in 64 character lines, just as it will be when printed onto paper, but you can't edit the text in this display.

RAMPrint's pull down options menu.



If you want to make any changes to the layout of your text you'll you'll have to go back into edit mode and 32 character display.

This is a bit of a bore. In practice I found that I could just go ahead and slap the text onto the screen and it would come out of the printer looking okay, but I still wanted to get a look at the text as it would appear on paper and this meant a lot of switching to and from Display mode. If you're used to the 64 column display of Tasword or The Writer this can seem a bit of a step backwards and a lot depends on whether you find the convenience of the interface/software combination an adequate compensation for the relatively unsophisticated editing facilities.

On the other hand you could easily use Tasword with RAM Print, and this will give you the best of both worlds.

Despite the shortcomings of RAM Write, the convenience of having so much crammed into the interface and the ease of use that RAM Print offers make the whole thing a godsend for someone like myself who is totally baffled by the intricacies of printers and control codes and the like. At last I've gotten my hands on a word processing set-up that you can just plug in and use straight away. Don't forget that a decent Centronics interface would cost about £30 anyway, and for just £35 RAM Print offers a Centronics interface, plus cable, joystick interface, a passable wordprocessor and 'user friendliness' that it's difficult to put a price on. In other words I liked it a lot. RAM Print is the most useful bit of hardware to come my way for ages.

RANDOM MEMORY

You may remember a few issues ago I suggested using a series of short subroutines, called in succession to produce an illustration. Now's the time to follow up that idea.

As 3-D pictures seem to be in vogue at the moment let's set about producing the graphics for a dungeon/passage-type adventure. We will need views into rooms, along passages, around corners, into dead-ends, through junctions etc. Although this may seem a little daunting it is in principle quite simple as all of these views can be made up of just a few, often repeated lines. Type in **Program 1**, keeping to the line numbers, and I'll show you what I mean. Now call a few lines with the command GOSUB (line number). In each case a few lines will appear on screen. Not very impressive you might think, but don't give up yet. These are only the picture elements. Add **Program 2** to the listing for Program 1 you have on board. Now set PAPER to 0 and INK to 7 — It'll look more "dungeony" in white on black background — then GOSUB to each of the new lines in turn, with a CLS between each call and you'll see how various combinations of those original picture elements can be joined to give a series of rooms, junctions, passages, turns etc.

PROGRAM 1

```
  10 PLOT 16,175: DRAW 48,-48: P
LOT 0,0: DRAW 64,32: RETURN
  11 PLOT 239,175: DRAW -48,-48:
 PLOT 255,0: DRAW -64,32: RETURN
  12 PLOT 15,175: DRAW 16,-16: D
RAW 0,-143: DRAW -31,-16: RETURN
  13 PLOT 240,175: DRAW -16,-16:
 DRAW 0,-143: DRAW 31,-16: RETUR
N
  14 PLOT 32,127: DRAW 191,0: PL
OT 32,32: DRAW 191,0: RETURN
  15 PLOT 223,127: DRAW -159,0:
DRAW 0,-95: DRAW 159,0: RETURN
  16 PLOT 32,127: DRAW 159,0: DR
AW 0,-95: DRAW -159,0: RETURN
  17 PLOT 223,127: DRAW -32,0: D
RAW 0,-95: DRAW 32,0: RETURN
  18 PLOT 32,127: DRAW 32,0: DRA
W 0,-95: DRAW -32,0: RETURN
  19 PLOT 65,126: DRAW 46,-46: P
LOT 190,126: DRAW -46,-46: PLOT
64,32: DRAW 46,23: PLOT 191,32:
DRAW -46,23: FOR n=16 TO 28 STEP
 4: PLOT 95+n,96-n: PLOT 160-n,9
6-n: PLOT n+95,n/2+48: PLOT 160-
n,n/2+48: NEXT n: RETURN
  20 PLOT 191,127: DRAW -127,0:
DRAW 0,-95: DRAW 127,0: DRAW 0,9
5: RETURN
  21 PLOT 32,159: DRAW 191,0: PL
OT 32,16: DRAW 191,0: RETURN
```

So far so good, but our adventurer would be rather restricted in his movements. Let's add some doors, arches, openings, holes, ledges, and a crevasse! While we're at it we might as well supply some flaming torches to through a glimmer of light on the proceedings. Add **Program 3**, noting that the capital letters are all udgs, and must be entered in the G mode. We'll also need to set these udgs so input the data from **Table A**, using
FOR f=USR "a" TO USR "p" + 7 : INPUT i : POKE f, i : NEXT f

Now you can add a lot more features to your basic rooms and passages. Try:
9999 LET i=2 : LET c=10 : GOSUB 36 : GOSUB 37 : GOSUB 3 : GOSUB 22 : GOSUB 26 : GOSUB 27 : GOSUB 28 : GOSUB 39 : GOSUB 41 : GOSUB 42 : GOSUB 44

You should get a view similar to that shown in **Fig. 1**. Note that the value of c determines the position of opening, arch or hole on the facing wall produced in lines 29-37. Also the value of i in lines 41-44 determines the INK colour of the handle of the torches. (The shimmering effect in the torches is caused by nothing more than the Spectrum's infamous "dot creep". There just had to be a use for it!) I'll leave you to experiment further — the function of each line is REMed in the listing — and to discover impossible combinations. (Like trying to include a crevasse in a room!)



Instant Graphics

It may be that the perfectionists amongst ZXC readers are not hapy with the way the picture is drawn before their very eyes. It may be that you would prefer the illustration to suddenly appear, complete, on screen. All that is needed for this effect is a combination of trickery and a short machine code routine. Use the following line to input these numbers: **33, 0, 88, 1, 192, 2, 22,**

PROGRAM 2

```
   1 GO SUB 12: GO SUB 13: GO SU
B 21: RETURN : REM room
   2 GO SUB 10: GO SUB 11: GO SU
B 19: RETURN : REM forward
   3 GO SUB 10: GO SUB 11: GO SU
B 20: RETURN : REM dead end
   4 GO SUB 10: GO SUB 15: GO SU
B 13: RETURN : REM right
   5 GO SUB 11: GO SUB 12: GO SU
B 16: RETURN : REM left
   6 GO SUB 12: GO SUB 13: GO SU
B 14: RETURN : REM right/left
   7 GO SUB 10: GO SUB 13: GO SU
B 17: GO SUB 19: RETURN : REM fo
rward/right
   8 GO SUB 11: GO SUB 12: GO SU
B 18: GO SUB 19: RETURN : REM fo
rward/left
   9 GO SUB 12: GO SUB 13: GO SU
B 18: GO SUB 19: GO SUB 17: RETU
RN : REM forward/left/right
```

PROGRAM 3

```
22)PRINT AT 13,7;"O": PLOT 64,96: DRAW -32,16: DRAW 0,-96: DRAW 32,16: DRAW 0,64: RETURN : REM left door
23 PRINT AT 8,4;"AB";AT 9,4;"■AB";AT 18,4;"■■CD";AT 19,4;"CD": FOR n=10 TO 17: PRINT AT n,4;"■■■■": NEXT n: RETURN : REM left opening
24 PRINT AT 8,4;"EFP";AT 9,4;"■■■P": RETURN : REM left arch
25 PRINT AT 13,24;"O": PLOT 191,32: DRAW 32,-16: DRAW 0,96: DRAW -32,-16: DRAW 0,-64: RETURN : REM right door
26 PRINT AT 8,26;"EF";AT 9,24;"EF■■";AT 18,24;"GH■■";AT 19,26;"GH": FOR n=10 TO 17: PRINT AT n,24;"■■■■": NEXT n: RETURN : REM right opening
27)PRINT AT 8,25;"OAB";AT 9,24;"O■■■": RETURN : REM right arch
28 PRINT AT 13,17;"o": PLOT 111,32: DRAW 33,0: DRAW 0,72: DRAW -33,0: DRAW 0,-72: RETURN : REM centre passage door
29 FOR n=9 TO 17: PRINT AT n,c;"■■■■": NEXT n: RETURN : REM centre passage opening
30 PRINT AT 7,c;" OP";AT 8,c;"O■■P": RETURN : REM centre passage high arch
31 PRINT AT 8,c;"EFAB": RETURN : REM centre passage low arch (centre)
32 PRINT AT 15,17;"o": PLOT 111,16: DRAW 33,0: DRAW 0,72: DRAW -33,0: DRAW 0,-72: RETURN : REM centre room door
33 FOR n=11 TO 19: PRINT AT n,c;"■■■■": NEXT n: RETURN : REM centre room opening
34 PRINT AT 9,c;" OP";AT 10,c;"O■■P": RETURN : REM centre room high arch
35 PRINT AT 10,c;"EFAB": RETURN : REM centre room low arch
36 FOR n=7 TO 9: PRINT AT n,c;"■■": NEXT n: RETURN : REM hole
37 PRINT AT 6,c;"OP": RETURN : REM hole arch
38 FOR n=96 TO 97: PLOT 28,n: DRAW 200,0: NEXT n: RETURN : REM room ledge
39)FOR n=96 TO 97: PLOT 60,n: DRAW 136,0: NEXT n: RETURN : REM passage ledge
40 FOR f=18 TO 20: PRINT AT f,8;"■■■■■■■■■■■■■■■■": NEXT f: PLOT 63,32: DRAW OVER 1;-48,-24: DRAW 224,0: DRAW OVER 1;-46,23: RETURN : REM crevass
41 PRINT AT 6,1; BRIGHT 1; INK 6;"KL";AT 7,1;"MN";AT 8,1; INK 1;"■■";AT 9,1;"■ ": RETURN : REM near left torch
42 PRINT AT 6,29; INK 6; BRIGHT 1;"KL";AT 7,29;"MN";AT 8,29; INK 1;"■■";AT 9,29;"■ ": RETURN : REM near right torch
43 PRINT AT 11,10; BRIGHT 1; INK 6;"I";AT 12,10; INK 1;"J": RETURN : REM far left torch
44 PRINT AT 11,21; BRIGHT 1; INK 6;"I";AT 12,21; INK 1;"J": RETURN : REM far right torch
```

A packed column this month with Clyde Bish revealing the minimal memory route to 3-D adventure graphics plus a useful 'Speedraw' routine.

7, 114, 35, 11, 120, 177, 32, 249, 201
FOR f=USR "t" TO USR "u" + 7 : INPUT i : POKE f, i : NEXT f
(If you want to use all the udgs you could move this code elsewhere). To call an illustration use a line such as **9999 CLS : INK O : GOSUB (each of the element subroutines) : RANDOMIZE USR "t" : INK 7**

This is how the trick works. INK is set to O — the same as PAPER — before calling the subroutines so they are drawn invisibly. Calling the machine code causes all the attribute file bytes to be altered to 7, i.e. white INK on black PAPER, so the picture appears more or less instantaneously. Finally INK is reset to 7 or you wouldn't see any subsequent printing! Note that the torch subroutines have to be called after the machine code or you'll get no colour. If you want of use, say, yellow INK on blue PAPER,POKE the attribute you require — in this case 6 + 1x8=14 — into address (start 7) as well as making the obvious changes to the INK number.

If you want your adventurer to have something to look at in the second it takes to produce the picture you could print a message after the CLS. The graphics won't affect it as long as you avoid areas which will be drawn on. Generally rows 0 to 5, columns 8 to 23 are safe, or you could use the edit lines with PRINT 0; "message".

Using different combinations of doors, openings, arches etc. you can produce an enormous variety of illustrations. (There are 15 possible combinations of torches for a start, even if you don't alter the value of i!)

I'll follow up this idea of picture elements further in a later article when we'll look at "Lords of Midnight"-type scenes, but now to a completely different solution to the problem of big pics in few bytes.

Speedraw

If we had machine code routines to plot, draw, block in areas of paper and fill areas with ink we could build up a picture on screen very quickly from a string of data if this included information for choosing the right routine. Ladies and Gentlemen, the Great Wizard of Exon proudly presents "Speedraw", a graphics utility to produce high resolution illustrations drawn at high speed and at an average cost of only 250 bytes per screen. (Less if you reuse parts of one picture in another).

For the technically minded, this is how its done. The secret is in the sequence of numbers the routine encounters. Lets call this the Drawcode. The value 255 (and to some extent O) is reserved for use by the driver. O on its own means return. The number of 225's at the beginning of each sequence determines which mode is selected. Exit from or jumping within a routine is controlled again by 255's being encountered. Figure 2 gives a flow chart for each mode.

Use
CLEAR 64890 : FOR f = 64891 TO 65367 : INPUT i : POKE f,i :

TABLE A

240	252	255	255	255	255	255	255
0	0	0	192	240	252	255	255
255	255	255	255	255	252	240	192
255	252	240	192	0	0	0	0
0	0	0	1	7	31	127	255
7	31	127	255	255	255	255	255
255	63	15	3	0	0	0	0
255	255	255	255	255	63	15	3
16	24	52	44	86	44	74	52
255	255	60	60	60	60	60	60
1	5	5	5	10	42	42	84
0	64	64	80	80	168	168	148
85	41	40	42	82	82	85	53
84	82	82	74	170	146	84	92
1	3	7	15	31	63	127	255
128	192	224	240	248	252	254	255

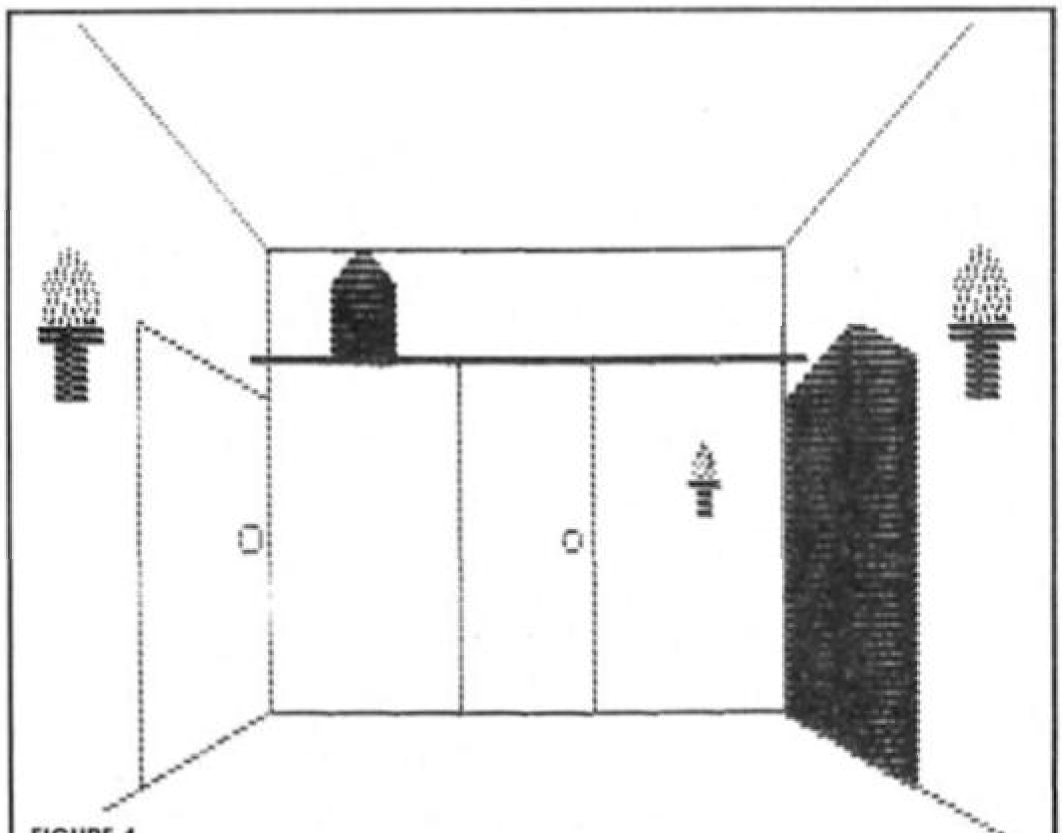


FIGURE 1

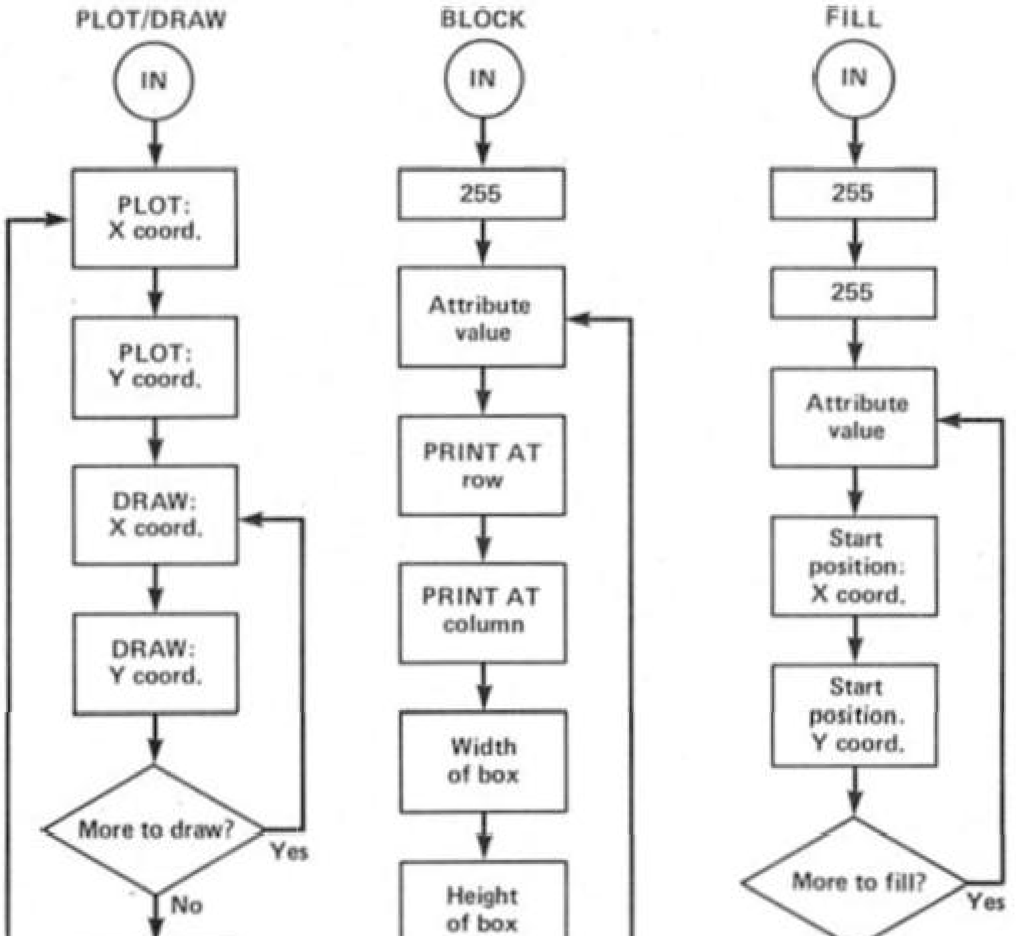


FIGURE 2: Control Data for Speedraw modes

PRINT f,i : NEXT f to enter the data for the machine code from **Table B,** and save with **SAVE "speedraw" CODE 64891, 477.**

Let's enter some example drawcodes to use each mode in turn.

Plot/Draw

This produces the outline drawing. Use the loader line **10 FOR f = 40000 TO 40021 : INPUT i : POKE f,i : NEXT f (RUN)** to enter the following sequence of numbers, pressing ENTER where there is a comma.

10, 10, 50, 10, 50, 60, 10, 60, 10, 10, 225, 100, 40, 160, 40, 130, 100, 100, 40, 255, 255, 0.

Now use RANDOMIZE USR 64891 to display a box and triangle to screen. If you compare the data with the flowchart you will notice (a) that this mode is called when NO 255 starts the sequence, (b) The first pair of numbers are the PLOT coordinates, subsequent pairs being **absolute** DRAW coordinates, i.e. you give the **actual** coordinates you want to draw to. No more messing about with positive/negative offsets; (c) a 255 starts a new PLOT position, (d) two 255's means return to the driver. Note that you cannot PLOT/DRAW to) or 255. You don't need to anyway as the routine draws a border around the picture area. (This becomes very important in the FILL mode).

Fill

This mode "inks in" the pixels within an area bounded by lines. As you will need an area to fill alter the FOR/TO numbers in the loader line to 40021 and 40030 and add the following data:

255, 255, 57, 11, 15, 58, 100, 42, 255, 0

RANDOMIZE USR 64891 to redraw the square and triangle and fill them with red and blue inks. Work out how it runs using the flowchart, remembering that the attribute values are ink + paper *8.

One or two points to note when using fill. (a) the area to be filled must be **completely** enclosed. The smallest gap and FILL leaks out. There may be "invisible" gaps along a line. You can close these with an extra PLOT-DRAW, but often just moving the start position of the fill will solve the problem. (b) in order to be very fast the fill is only semi-intelligent, so you need to have the start position against a left or right boundary. You will also need to have more than one start point to fill shadowed areas (e.g. filling a ring), and the full may not spread to very narrow areas. *Either complete these with* PLOT-DRAW or don't have narrow areas! Finally it is just possible to crash the routine so save the drawcode before trying it out.

TABLE B

17	34	255	26	103	19	26	111	32	220	225	201	125	230	192	31
6	4	197	19	26	71	19	26	55	31	15	173	230	248	173	87
79	213	205	21	254	209	193	16	124	7	7	7	173	230	199	173
241	17	63	156	19	26	254	0	7	7	95	124	230	7	71	4
200	254	255	194	234	253	19	26	62	254	15	16	253	71	26	160
254	255	194	196	253	19	26	50	168	47	18	201	237	75	125	92
143	92	19	26	111	19	26	103	197	17	0	255	120	130	71	254
213	34	125	92	205	151	254	209	255	40	114	254	176	40	110	197
19	26	222	255	32	232	195	151	205	170	34	60	71	197	126	7
253	26	50	45	255	19	26	50	16	253	193	15	56	14	13	16
48	255	19	26	50	47	255	19	250	125	43	230	31	126	40	4
26	50	50	255	19	26	50	51	6	8	24	239	8	62	9	144
255	213	205	44	255	209	19	26	71	8	7	48	17	12	29	40
222	255	32	221	195	151	253	26	67	16	247	35	125	230	31	126
103	19	26	71	62	176	152	111	40	58	6	8	24	236	30	255
19	26	71	19	213	26	87	62	12	227	105	227	8	253	126	87
176	154	79	205	21	254	209	19	15	15	47	230	1	79	8	177
26	222	255	32	236	19	26	222	28	16	17	119	229	213	205	219
255	32	220	195	151	253	1	0	11	209	225	35	125	230	31	40
1	1	197	205	111	254	193	17	16	6	8	126	7	48	232	15
1	1	120	148	48	4	21	21	7	16	253	119	213	205	219	11
237	68	71	121	149	48	4	29	209	193	24	136	193	122	254	1
29	237	68	79	176	200	121	184	200	17	0	1	193	24	136	0
229	98	107	34	19	254	46	0	175	255	175	255	0	0	0	0
56	4	101	107	72	71	34	17	175	62	32	1	15	10	17	12
254	96	120	203	63	111	125	129	6	33	223	87	213	17	32	0
56	3	184	56	8	144	111	237	4	25	16	253	65	4	35	16
91	19	254	24	5	111	237	91	253	209	66	229	72	67	119	35
17	254	227	124	130	103	125	131	16	252	225	213	17	32	0	25
111	197	205	111	254	193	227	37	209	65	16	239	201			

(Of course, once you have it working correctly it won't crash when in use in a program).

Block

This effectively "papers in" a block of character squares. Alter the FOR/TO numbers to 40000 and 40007 and enter the following sequence, which produces a blue block, 10 by 5 with top left corner at row 15, column 1 255, 14, 15, 1, 10, 5, 255, 0

As before RANDOMIZE USR 64891 will display it. As it is the attribute number which is used this mode can be used to change the ink within an area without affecting the paper, and so alter just the outline colour.

If you want to see a large picture drawn with Speedraw use the loader line (with FOR/TO set to 40000 and 40250) to enter the data from **Table C.** RANDOMIZE USR 64891 will quickly display a Hobbitish path in the Misty Mountains leading to the Golden Key that you don't need anyway (Oops — Sorry!). If you look carefully you'll see that the sky top left is BLOCKed, not FILLed. this is so you can FILL the moon. (Remember you can't have INK on INK).

When using Speedraw in your own programs you'll need to set the data start register to the beginning of the drawcode you want it to read. Do this with the subroutine:

1 RANDOMIZE a: POKE 64917, PEEK 23670: POKE 64918, PEEK 23671: RANDOMIZE USR 64891: RETURN having first set variable a to the data start address **less 1** of the drawcode you wish to use.

You have probably worked out by now that to produce an accurate drawcode will take a lot of working out on pixel paper. This cannot be avoided, but I can offer you some help in the form of an Editor/Assembler which will find and help you correct the errors. The listing is too long to include here, but if you're interested in finding out more about it write to me care of ZX Computing enclosing an SAE.

Next time we'll return to the picture element idea and produce those promised "Lords of Midnight"-type landscapes at only 30 bytes each.

'Til then, happy drawing!

TABLE C

68	159	115	159	115	158	56	158	255	175	71	154	22	255	183	39
56	159	67	159	71	174	255	56	163	17	255	83	174	80	160	255
159	38	52	44	47	55	44	76	143	174	128	128	255	193	174	160
43	74	34	76	33	88	30	127	112	255	220	174	200	88	255	254
24	144	23	152	22	160	19	164	168	247	152	232	64	255	39	167
15	165	10	164	7	159	1	255	33	166	29	162	27	156	29	149
124	174	112	152	109	144	112	139	33	145	39	144	36	150	35	156
120	132	127	127	159	111	199	87	36	162	39	167	36	166	255	1
215	77	231	63	247	39	254	24	174	4	170	17	142	31	123	40
255	110	152	106	148	104	143	107	115	47	112	1	112	255	47	174
135	112	128	144	104	157	93	176	47	111	255	48	114	48	130	50
72	182	56	184	39	184	31	176	130	50	120	49	120	49	130	255
15	168	1	255	84	157	57	45	255	255	9	0	0	6	8	255
255	103	141	74	34	75	33	103	255	255	8	1	119	48	1	1
141	255	119	119	96	29	255	143	49	48	174	14	28	156	50	169
103	126	25	255	167	79	146	23	1	255	0					

THE FUTURE IS HERE!

What better way to plan your adventures for 1987 than mark them on a stunning Level 9 year planner. Fifty ZX readers can rhyme their way to a prize.

"There once was man called Clive,
Who one morning brought the Spectrum alive,
He thought 'Oh well, I'll invent the QL',
And before lunch I'll create the C5."



One of four Godfrey Dowson illustrations that feature on the year planner

This shining example of how to write a limerick (okay there may be a few poetic bugs in it) should give you some pointers towards creating your own rhyming masterpiece. To adorn your wall with a Level 9 year planner all you have to do is complete the limerick below. Remember the last line must rhyme with the first two and the third and fourth with each other. Getting the right rhythm is important and prizes will go to those who have thought up the most apt and humorous ending.

The wall chart is fully laminated and washable and comes with a fibretip pen to pencil in your important engagements. Apart from being extremely useful the planner is decorated with four superb full colour illustrations by Godfrey Dowson.

Level 9 Yearplanner Competition

A spokesman for Level 9,
said adventures are truly divine,

____

____

____

Name: ____________________

Address: ____________________

Post to Level 9 Yearplanner Competition, ZX Computing Monthly, No. 1 Golden Square, London W1R 3AB.

The limerick

Time to get into limerick mode. The first two lines are...
A spokesman for Level 9,
said adventures are truly divine,

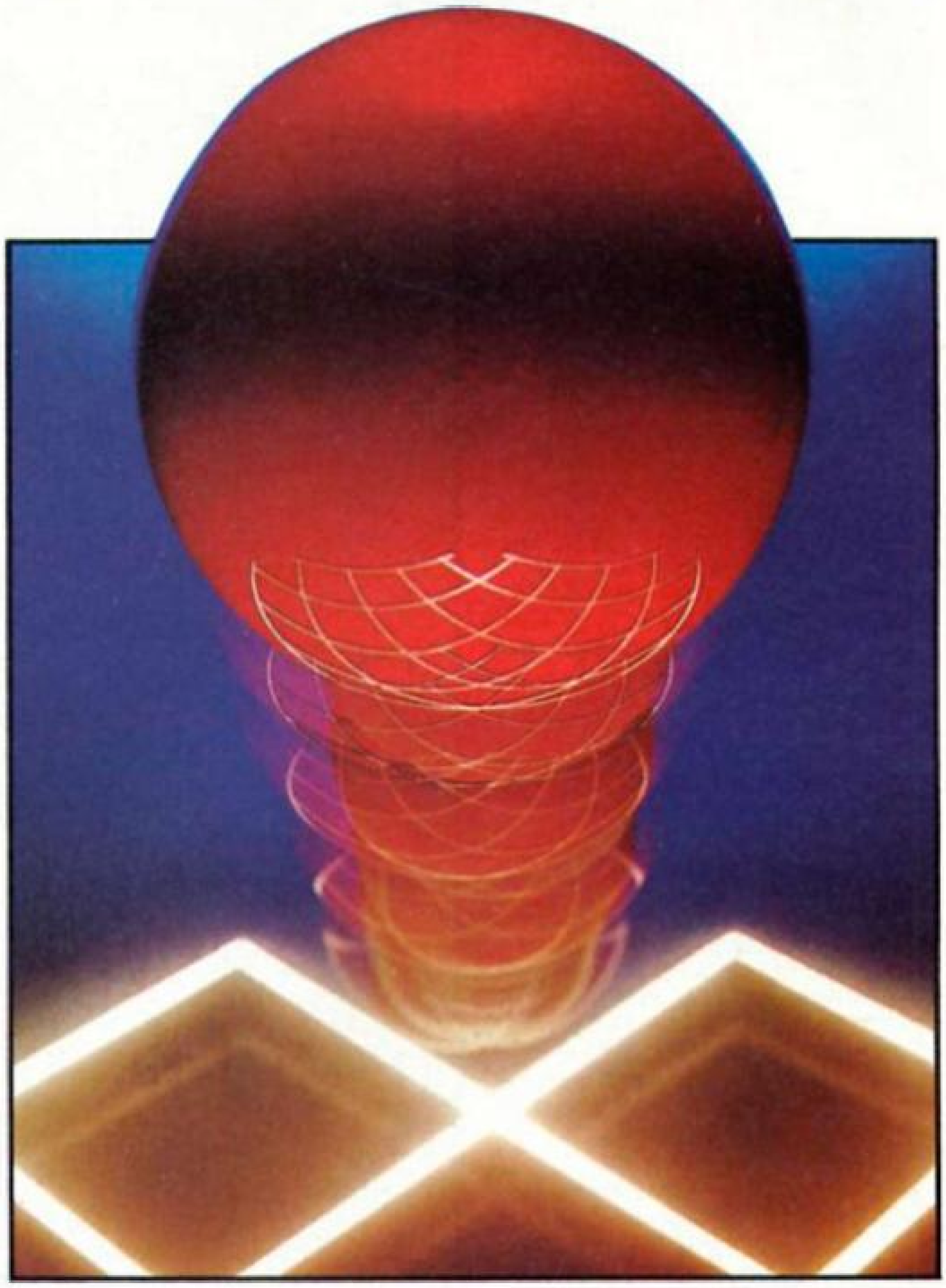
Now it's over to you to complete the last three lines. The competition is open to all ZX readers apart from employees of Argus Specialist Publications, Alabaster Passmore and Level 9. The closing date is December 5th 1986. Please remember to write your three lines on the back of your entry envelope.

Send your entry to Level 9 Yearplanner Competition, ZX Computing Monthly, No. 1 Golden Square, London W1R 3AB.

REVOLUTION

Masterly graphics, infuriating puzzles — all you ever wanted from a bouncing ball game.

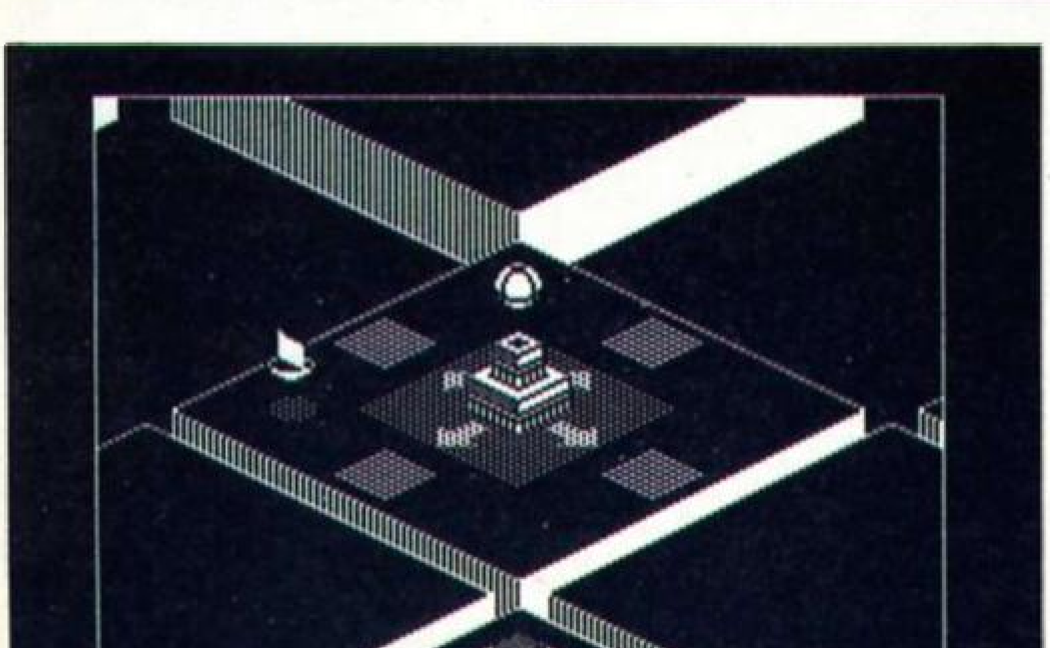
Revolution
Vortex/U.S. Gold
£9.99



Costa Panayi, author of many of Vortex Software's previous hits, has made a change in direction after his recent games Alien Highway and Highway Encounter and come up with another winner.

Revolution is one of those games which is quite simple to play (in principle anyway), but very difficult to master and absolutely infuriating when you get into the later levels of the game where you can see what you want to do but can't quite pull it off in time.

The game is played on a multi-tiered structure consisting of eight levels, and your task is to start on the bottom level and work your way to the very top. Each level is made up of an assortment of platforms, all on the same level but separated by gaps which have to be avoided. These platforms are arranged in a roughly square arrangement (which changes in each game) with the occasional empty position which you might find yourself plummetting into if you look before you leap (each of these levels is a bit like one of those sliding square puzzles in which you can move the squares around to try and form a picture or sequence of numbers).



You control a ball which can be bounced around the platforms and in order to complete a level and progress to the next you have to solve four puzzles. Scattered around each level you will find four platforms which carry various brick-like structures and shaded pathways. On each of these platforms there are two grey blocks which turn white when you bounce into them and deactivate them. Each brick will remain deactivated for just a few seconds before reactivating and turning grey again, but if you can get both bricks deactivated at the same time then they will both vanish — problem solved ▶▶▶

SPECTRUM GAME REVIEW

REVOLUTION

and you can move on to the next problem on that level.

Unfortunately you've also got to contend with the shaded pathways on these platforms, and these can have different effects varying from just killing your bouncing or grabbing the ball and shooting it over the edge of the platform into oblivion.

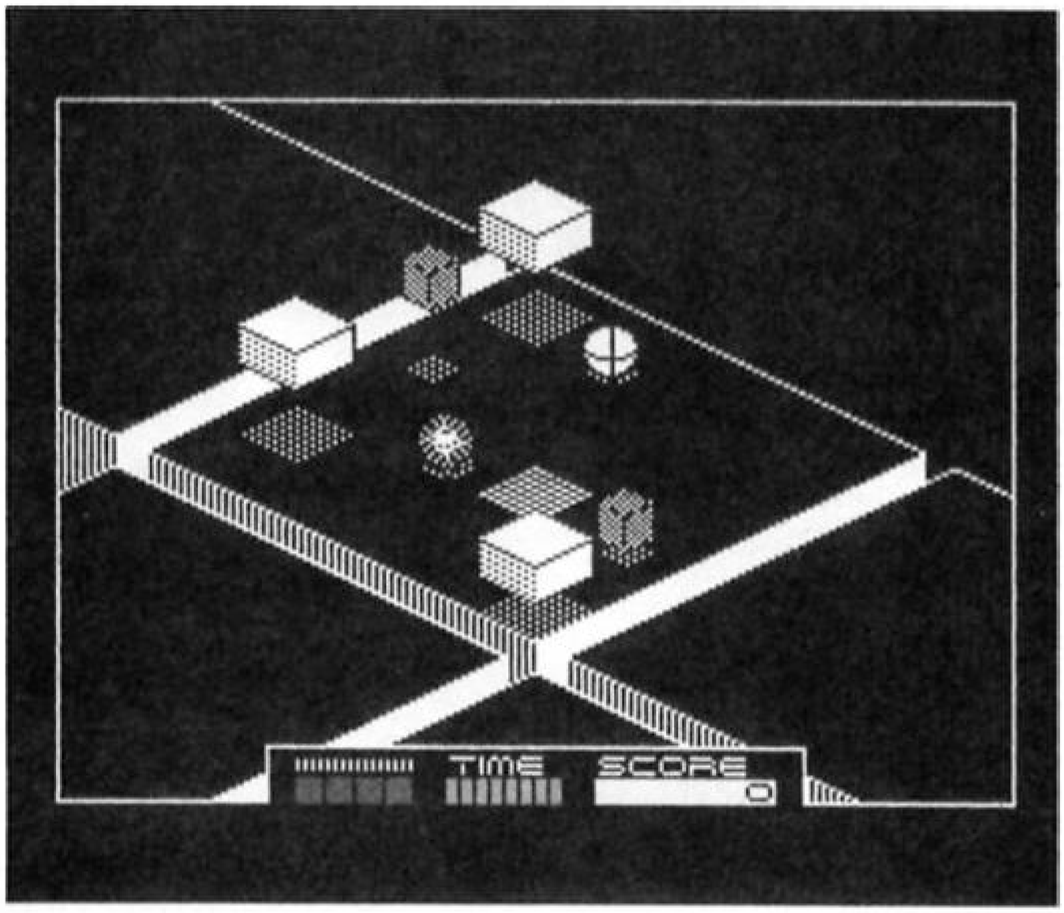
At the start of each level you are shown your position within the overall structure, as well as a map of the particular level that your are on. This map highlights both your own starting position and the position of the puzzles on that level so you've got a few tips to start off with. When you see this map it's also a good idea to make a mental note of the empty positions, as I found that I often went bouncing from one screen to the next only to find myself bouncing into nowhere. You only get five lives (though there are additional ones for completing levels) so you can't afford to do that too often.

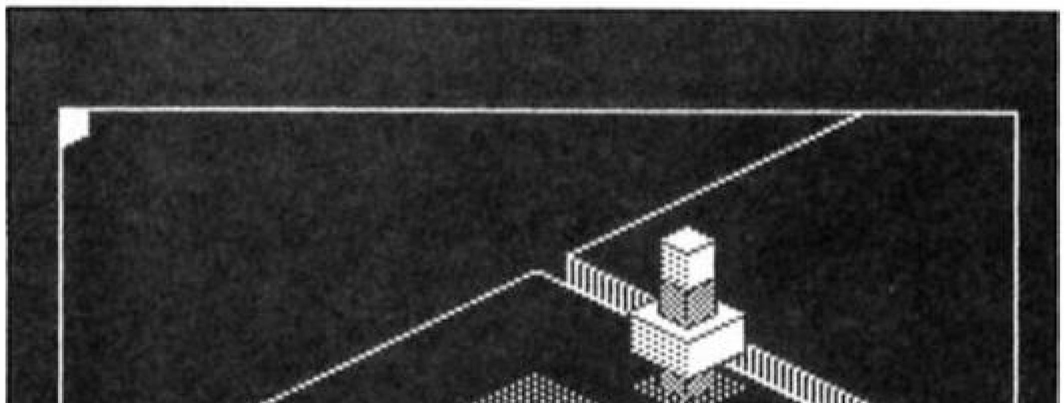
Controlling the ball is an art all in itself. It can move in any direction that a joystick handle can move in (a joystick is almost compulsory for this game I should think), and responds very smoothly to joystick control but judging the height and distance of your bouncing is a skill that will take a lot of refining, especially on some of the trickier puzzles where you've got to squeeze the ball through some quite tight spots.

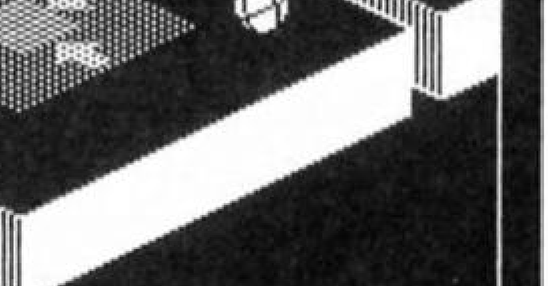
There are four different heights of bounce, controlled by using the fire button, or you can kill the bounce altogether and just roll along the surface of the platforms. There is also a variety of flying objects which patrol each level and though few of these are deadly you can find yourself accidentally colliding with one and bouncing off in a totally unexpected direction.

The graphics are mostly in just two colours to avoid attribute clashes, but as with Vortex's Highway games the graphic style is quite distinctive, making good use of fine shading effects to create clear and detailed pictures. But where I found the Highway games to be a bit too fast and furious for my gin-addled reflexes, Revolution is the sort of game that each player can play at his or her own pace. If you want to go bouncing around the platforms like a loony you're quite free to do so, but if you're content to take things a little bit slower and just roll slowly around while you try and figure out which way is up you can do that too.

At £9.99 Revolution is a bit pricey, but it's more likely to provide a long term challenge than most other arcade games around at the moment.







Revolutionaries

ZX talks to Salford based software house Vortex about the intricacies of Revolution.

Revolution, a complex 3-D brainteaser is a real test for puzzle solvers as the game layout subtley changes with each new game. Luke Andrews of Vortex explained the evolution of Revolution.

"Firstly it was essential to create the game in 3-D. It offers an attractive illusion and greater scope for complicated gameplay. The bouncing ball idea was really a progression from Highway Encounter but we wanted to do something completely different with it.

"The idea had been around in Costa's head for over a year until he got the confidence to create such a game. What happens is that Costa comes up with an overall graphic treatment and we discuss ideas and ways of creating improved gameplay.

On the level

"The levels idea came at a later stage. We knew the bouncing ball was going to be the main character but we wanted an alternative to a long drawn out game on one level and something different from the usual map. With the idea of the levels we solved both problems. This way you could identify where you are and the overall layout very simply."

One of the infuriating aspects of puzzle based games is laboriously working through the puzzles you've already solved to get back to the position you reached in your previous forays. In Revolution this problem doesn't arise.

"There are eight layouts in the stack and the chances of meeting those layouts in the same order twice are 1 in 10,000. There's also a checkerboard layout that you encounter in the later levels. The random element is built-in in two ways. Firstly the puzzles are mixed up in the various levels and secondly you may find one puzzle on an early level and again later but you will have less time to come up with the solution.

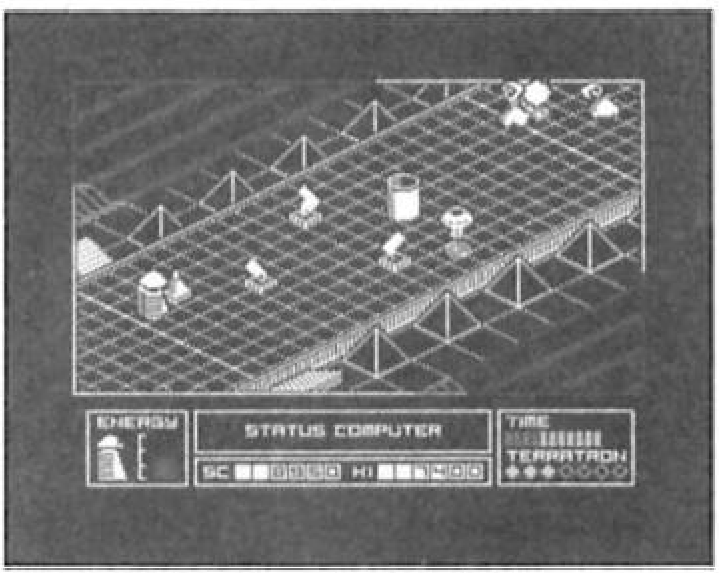
Complications

"Programming games like this presents immense difficulties and although most of them are ironed out in the early stages. One problem was that when the ball was perched on an object

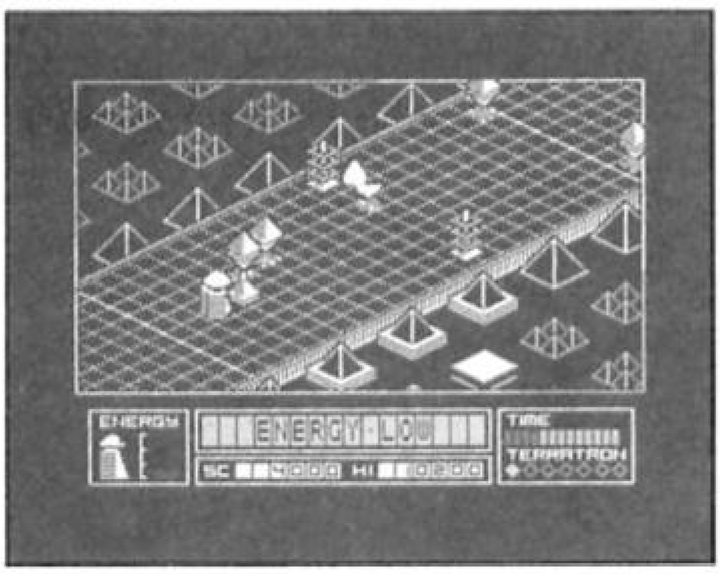


Vortex: Crete Panayi, Luke Andrews and Costa Panayi

it would just fall off rather than roll or pop up. It was solved, but getting the ball to look convincing and realistic was perhaps the hardest part.

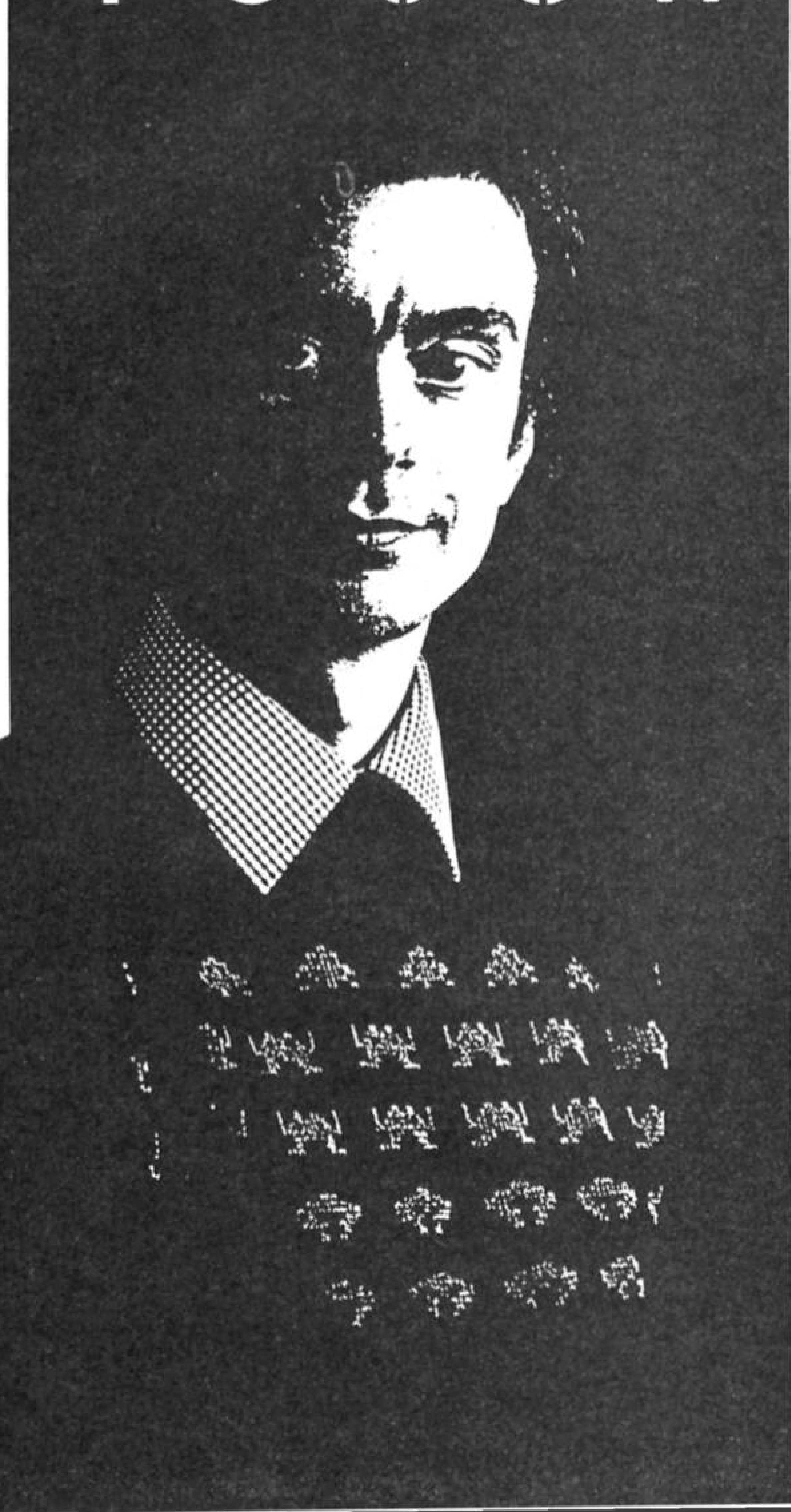


Vortex's Alien Highway: Scheduled for budget re-release on U.S. Gold's Americana label



"The idea of touching the two blocks to solve the puzzle came about half way through the development. The three of us have been together for about four years now and when an idea comes up that we like we all get enthusiastic. If any of us hesitate then we ditch it. With the blocks we knew it was the vital ingredient we needed."

THE PROFESSIONAL TOUCH



This month, Hewson's Steve Turner has some advice on how to Supercharge Your Spectrum.

In the past two years the standard of Spectrum programs has increased dramatically. To compete in today's market you have to push the Spectrum to its performance limits. This month I'll look at a few tricks of the trade that can be used to realise the full potential of the 48K machine.

Increasing user RAM

To use all of the RAM apart from the screen and attribute maps you must shut off the Spectrum ROM. Every 50th of a second a hardware timer causes an interrupt. This has the effect of giving control to the ROM keyboard scan routine. This interrupt needs to be switched off with a disable interrupt instruction. Of course no ROM routines can be used or the Spectrum ROM variable area may be corrupted (I have always avoided using ROM routines. It is usually much better to design your own routines that do exactly what is required).

You can also use more RAM than there is! This is achieved by using areas twice. Routines and data only used at the start of the program can be overlaid with other routines loading over them. In AVALON for example, the code for keyboard choice, displaying the pretty border and initialising everything is overwritten by the main game.

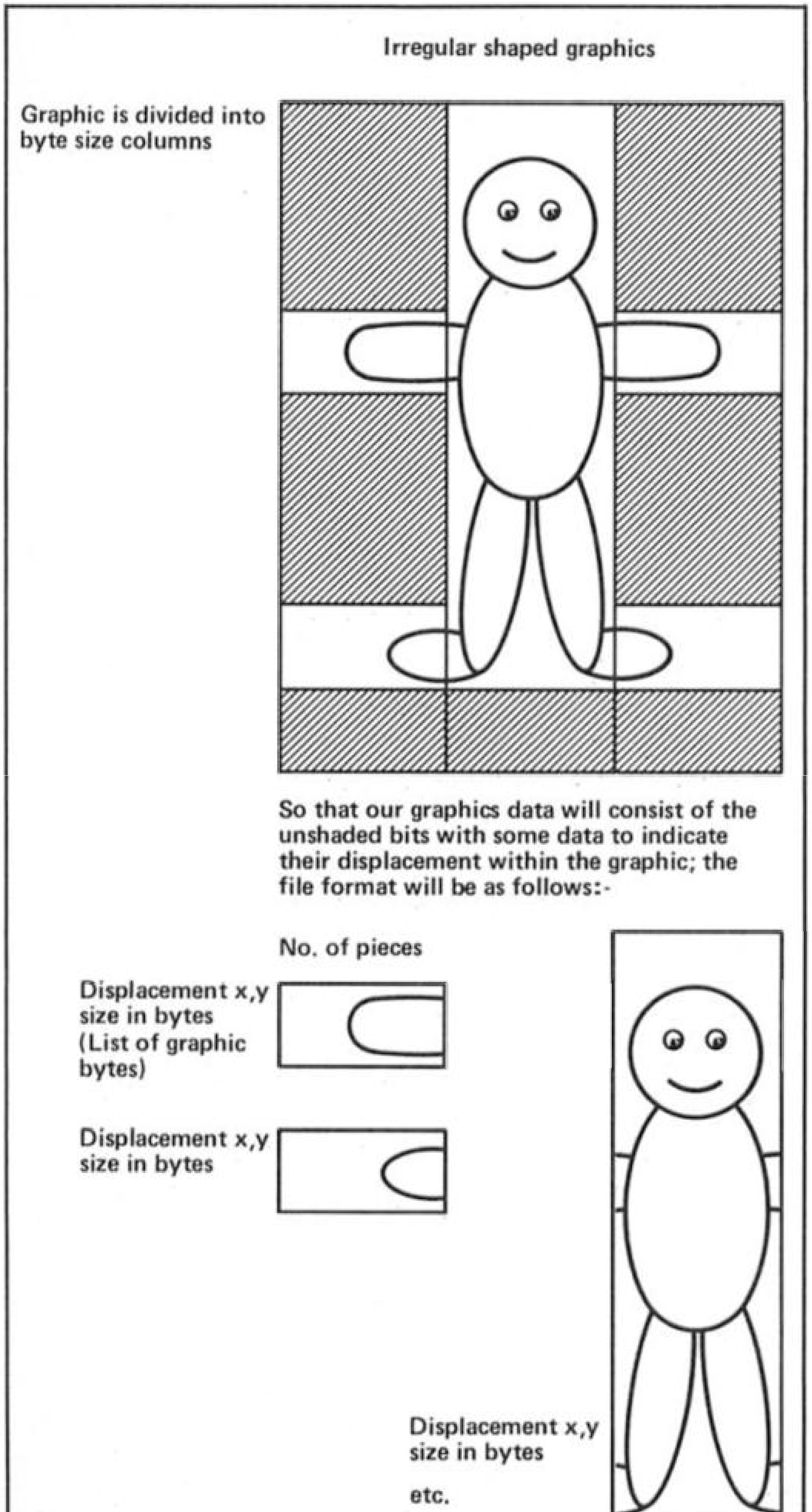
Both of these methods are best implemented in the final stages of testing. Once applied the game is in a 'live' state and can only be started once. It is also extremely difficult to get back to BASIC.

Reducing the size of your code

It is best to write code using as few bytes as possible from the outset of coding. This has to be balanced with clarity and execution speed requirements.

Here is a list of tips to reduce the size of your code.

1. Use **XOR A** instead of **LD A,0**
2. Use **AND A** instead of **CP A,0**
3. Structure your programs to avoid unnecessary repetition.
4. Make large macros into subroutines.
5. To increment counts etc make use of **INC (HL).**
6. Use block moves and 'compares' when appropriate.
7. Use table-driven processing so that each routine performs a greater range of tasks. For



instance, instead of coding a movement routine for each object in a game tabulate the movement characteristics, then write a general routine that accesses the table and produces a different result for each entry.

Reducing the size of data

When designing data tables bear two things in mind; how often will the data be used and how tightly can it be packed?

Generally, packing data means a slower access time so often a compromise is necessary. Here are the techniques I use.

1. Use variable length data records where you can.
2. Use ROM or Machine code as table data as "free" data. Background wall objects in AVALON were chosen and placed using ROM as a data table, and I adjusted the data to be in the required ranges. It is similar to using a random value except if you get the same value whenever you want it.
3. Design the data so that as much information as possible can be deduced from the record number. In AVALON room colours were a function of room number.
4. Do not include the record number in the record. The program either knows it or can deduce it from the record's position.
5. Look for common or repeated data and group these records, specifying the common ones.
6. Pack maps or suitable data and unpack them into a buffer for fast access.

Reducing RAM usage of graphics

Graphics data usually takes up a major part of the RAM in a state-of-the-art arcade game. The graphics, when used, may have to be accessed many times a second, so we must decide which graphics we can pack and which have to be in a format suitable for fast plotting.

One of the easiest ways of getting more graphics in the program is to use custom character sets as building blocks. Large graphics can then be constructed from the characters. A data record for each object specifies its dimensions and which characters it is made up of. If most of the characters are used many times the RAM savings are considerable.

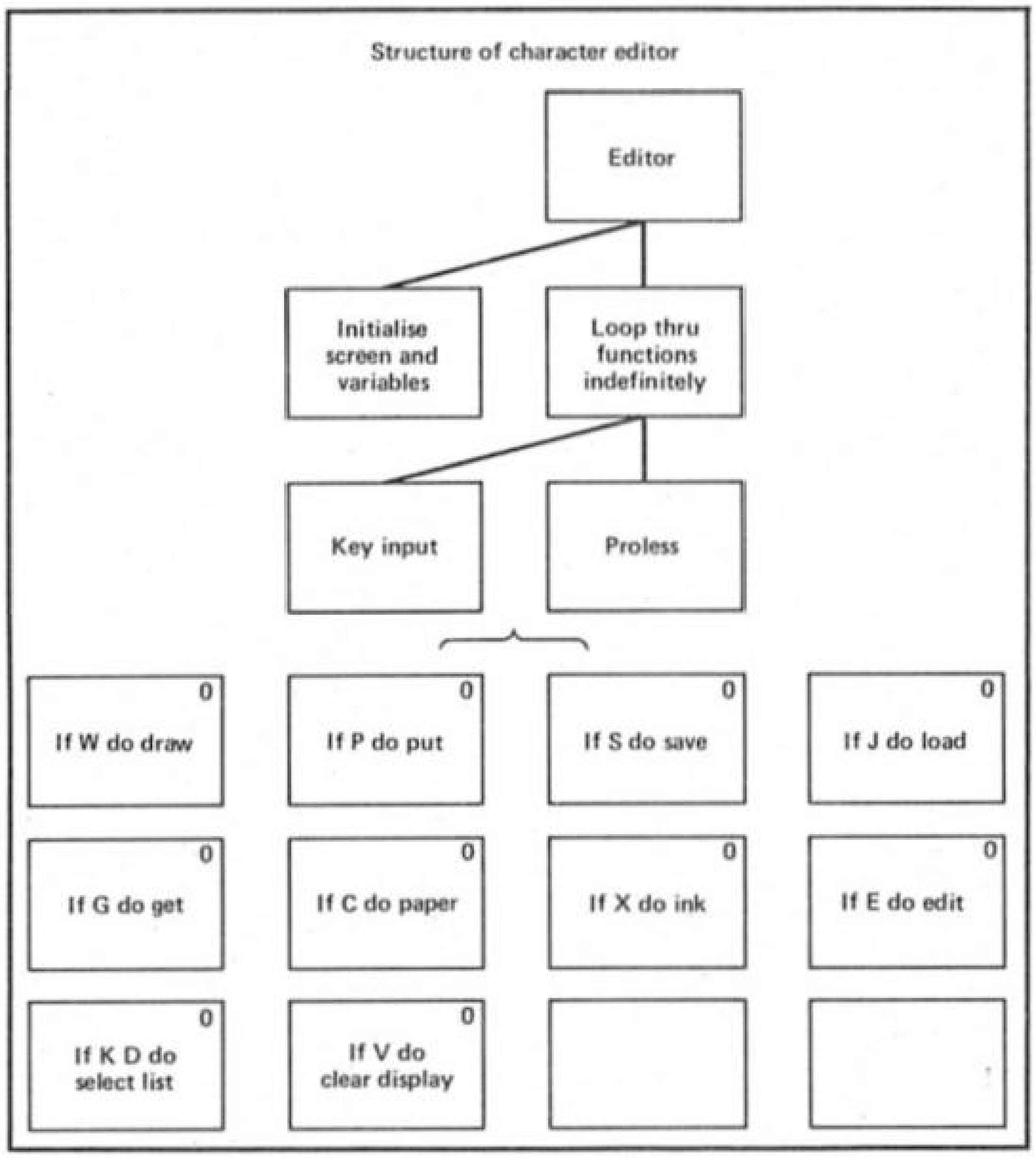
An extension of this idea is to use a building block larger than a single character. This system could be used to program a game like Knightlore and its derivatives. Each 3D block could be one of the basic building graphics. By plotting them on top of one another, walls, towers and arches can be constructed.

The choice of the shape and the size of building block depends on the style of the game. I have used variable size blocks which lead to a very flexible system. You can even build macro blocks out of small blocks using even less RAM, though there is a price to pay. The CPU has more work to do to decode the data and build the graphics. This is why Quazatron only scrolls slowly. The poor CPU is busy constructing the bit of the picture moving into view.

Another way to optimise space used by graphics is to design a graphics format that caters for odd shaped objects. This also saves the CPU time as it does not have to waste time plotting blanks.

Increasing execution speed

The place to concentrate on



speed optimisation is in the innermost loops of the program. One instruction saved here can be the equivalent of thousands of instructions saved elsewhere if you can identify the routines that are used the most. It is sometimes useful to get an idea of how long a routine is.

To do this code a **HALT** instruction and set the border to a different colour. Restore the border to its original colour after the timed routine has finished, then repeatedly call the routine. In most cases the routine will already be in some sort of loop and the program can be run as normal. The size of the coloured stripe on the upper border gives an indication of execution time. If the routine executes for more than a 50th of a second the whole border will flash the chosen colour.

Usually the place where speed is most required is in the graphics plotting routines. In my games the CPU spends about 90 per cent of its time driving the picture on the screen. The design of the graphics routines makes or breaks a Spectrum game, so I will delve into the theory of animation so that the problems can be identified.

Animation

Animation is moving pictures. It is achieved by replacing one picture with another in quick succession, the more frames per second the smoother the animation. This is done by building up a new picture or part of a picture off screen. Background objects are drawn first, foreground objects last. Finally this new frame is copied to the screen.

The speed is affected by the size of the buffer or buffers that are built and copied. This is why so many games reduce the size of the moving screen area. The cleverest games manage to maintain a large moving area by just plotting the parts of the screen where changes occur.

The routines that clear, build and copy the buffers are worth optimising. This is where the craftiest techniques are employed. Let us examine the various methods applied to clearing a buffer to zeroes.

Using a LDIR instruction to copy the buffer to itself is a good method but not the fastest. It takes 21 cycles for each of the bytes to be cleared, plus setting up overheads. If interrupts are disabled the stack pointer can be used to PUSH two zeroes a time into the buffer. It only takes 11 cycles for each two bytes cleared plus 27 cycles as a loop control. If several PUSH instructions are coded in a list then the 27 cycle loop control does not have to execute for each two bytes cleared. The longer the list of PUSH instructions the faster the routine. Space considerations usually force a compromise.

```
  10 CLEAR 44999
  20 BORDER 6: PAPER 6: INK 0: CLS
  30 LET a$="": LET c=0: LET q=c: LET x=c: LET y=c: LET a=c: LET b=c: LET d=6:
LET e=c: LET f=c: LET g=c: LET h=1
  40 PRINT AT 0,4;"CHARACTER EDITOR   CHAR LIST"
  50 GO SUB 100
  60 PRINT AT 3,11;"W=DRAW";AT 4,11;"G=GET P=PUT";AT 5,11;"E=EDIT";AT 6,11;"S=SA
VE";AT 7,11;"J=LOAD";AT 8,11;"K=LIST";AT 9,11;"X=INK C=PAP";AT 10,11;"V=CLEAR PI
C"
  70 PLOT 15,160: DRAW 65,0: DRAW 0,-65: DRAW -65,0: DRAW 0,65
  80 PLOT 0,80: DRAW 183,0
  90 PLOT 183,0: DRAW 0,175
  95 GO SUB 9900
 100 REM function loop
 110 IF a$="W" OR A$="w" THEN  GO SUB 400: GO TO 100
 120 IF a$="p" OR a$="P" THEN  GO SUB 4500: GO TO 100
 130 IF a$="s" OR a$="S" THEN  GO SUB 5000: GO TO 100
 135 IF a$="j" OR a$="J" THEN  GO SUB 6000: GO TO 100
 140 IF a$="G" OR a$="g" THEN  GO SUB 4000: GO TO 100
 150 IF a$="C" OR a$="c" THEN  GO SUB 8000: GO TO 100
 160 IF a$="x" OR a$="X" THEN  GO SUB 9000: GO TO 100
 170 IF a$="E" OR a$="e" THEN  GO SUB 300: GO TO 100
 175 IF a$="k" OR a$="K" THEN  GO SUB 500: GO TO 100
 177 IF a$="V" OR a$="v" THEN  GO SUB 3000: GO TO 100
 180 GO SUB 300: GO TO 100
 300 REM EDIT CHARACTER
 305 LET I=X+(32*Y)+22594
 306 LET j=120: IF PEEK i=0 THEN  LET j=16
 307 POKE I,J:
 310 GO SUB 9900
 320 IF a$="0" OR a$=" " THEN  PLOT  OVER 1;96+x,159-y: GO TO 370
 336 LET j=0: IF PEEK i=120 THEN  LET j=48
 337 POKE I,J
 338 IF CODE a$>58 THEN  RETURN
 340 LET dx=dx+x: IF dx>=0 AND dx<8 THEN  LET x=dx
 350 LET dy=dy+y: IF dy>=0 AND dy<8 THEN  LET y=dy
 360 GO TO 300
 370 LET j=16: IF PEEK i=16 THEN  LET j=120
 375 IF INKEY$<>"" THEN  GO TO 375
 380 GO TO 307
 400 REM DRAWING
 405 LET I=A+(32*B)+22912
 410 POKE I,128+PEEK I
 420 GO SUB 9900
 450 POKE I,E+(D*8)
 460 IF CODE a$>58 THEN  RETURN
 470 IF a$="0" OR a$=" " THEN : LET k=45000+((c+q)*8): FOR l=0 TO 7: POKE USR "u
"+l,PEEK (k+l): NEXT l: PRINT AT 12+b,a; INK e; PAPER d;CHR$ 164: GO TO 400
 480 LET dx=dx+a: IF dx>=0 AND dx<22 THEN  LET a=dx
 490 LET dy=dy+b: IF dy>=0 AND dy<10 THEN  LET b=dy
 495 GO TO 400
 500 REM select char
 510 POKE 22583+(32*c),240
 520 GO SUB 9900
 530 POKE 22583+(32*c),120
 540 IF CODE a$>58 THEN  RETURN
 545 POKE 22583+(32*c),48
 550 LET dy=dy+c: IF dy>=0 AND dy<20 THEN  LET c=dy
```

```
 560 IF c=dw THEN  GO TO 500
 570 IF c=0 AND q<>0 THEN  LET c=19: LET q=q-20: GO SUB 1000: GO TO 500
 580 IF c=19 AND q<1000 THEN  LET c=0: LET q=q+20: GO SUB 1000: GO TO 500
 590 GO TO 500
1000 LET k=45000+(q*8):
1010 FOR r=0 TO 19
1020 FOR l=0 TO 7: POKE USR "u"+l,PEEK (k+l): NEXT l: PRINT AT 1+r,23;CHR$ 164;"
 ";q+r;"    "
1025 LET k=k+8: NEXT r
1027 POKE 22583+(32*c),120
1030 RETURN
3000 REM CLS BOTTOM AREA
3005 INPUT "CONFIRM CLS Y=YES ";A$
3010 IF A$="Y" OR A$="y" THEN  FOR L=11 TO 21: PRINT AT L,0,"        ": NEXT L
3020 LET a$="u": RETURN
4000 REM get char from list
4010 LET K=((q+C)*8)+45000
4020 LET N=16460
4030 FOR L=1 TO 8
4035 LET O=PEEK K: LET K=K+1
4037 POKE N,O: LET N=N+256
4040 FOR M=1 TO 8
4045 LET P=O: LET O=INT (O/2): LET P=P-(O*2)
4050 LET J=0: IF P=0 THEN  LET J=48
4055 POKE (22570-M)+(32*L),J
4060 NEXT M
4070 NEXT L
4075 LET A$="E"
4080 RETURN
4500 REM copy char to list
4510 LET K=((q+C)*8)+45000
4520 LET N=16460
4540 FOR L=0 TO 7: POKE k+l,PEEK (n+(256*l))
4550 POKE USR "u"+l,PEEK (k+l): NEXT L
4560 PRINT AT 1+c,23;CHR$ 164
4565 POKE 22583+(32*c),120
4570 LET A$="E": RETURN
5000 REM SAVE
5005 INPUT "NAME ",A$;" LAST NO ";O
5010 SAVE A$CODE 45000,O*8
5021 GO SUB 9900
5030 RETURN
6000 REM LOAD
6005 INPUT "NAME ",A$;AT 0,0;"FIRST CHARACTER NO ";O
6007 LOAD A$CODE 45000+(O*8)
6010 GO SUB 1000
6020 LET a$="e": RETURN
8000 INPUT "PAPER=",D: IF D>7 THEN  GO TO 8000
8010 LET a$="w": RETURN
9000 INPUT "INK=",E: IF E>7 THEN  GO TO 9000
9010 LET a$="w": RETURN
9900 LET a$=INKEY$: IF a$="" THEN  GO TO 9900
9907 LET dx=0: LET dy=0
9910 IF a$="5" THEN  LET dx=-1
9920 IF a$="8" THEN  LET dx=1
9930 IF a$="6" THEN  LET dy=1
9940 IF a$="7" THEN  LET dy=-1
9999 RETURN
```

If interrupts are enabled slower methods have to be used such as listing LD (HL),A and INC HL instructions or LDI instructions.

Fast and slow

It is worth designing your program so that heavily used data and routines do not occupy the lower 16k of RAM. This is because all RAM from 4000h to 7FFFh is considered by the hardware as the screen bank of memory. Every time the video hardware requires a byte to put out to the TV, and the CPU is fetching data from the lower RAM, the CPU is temporarily stopped in mid instruction. This can make it 30 per cent slower.

Character Generator

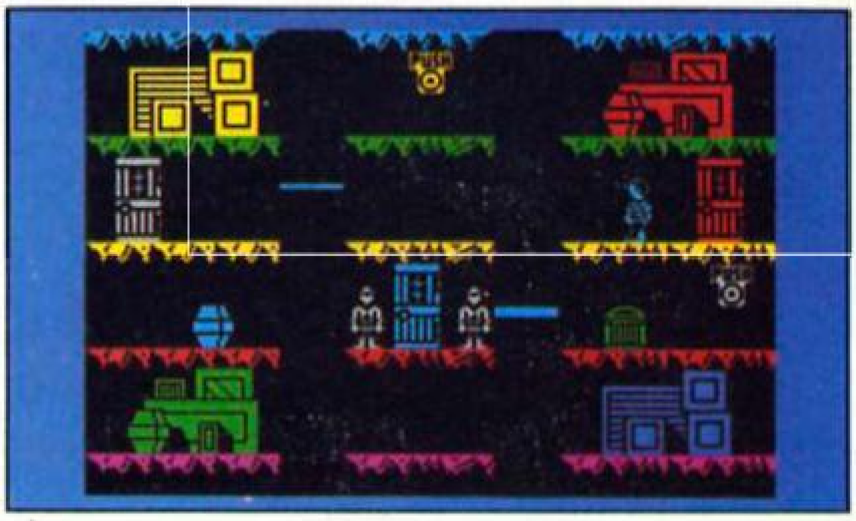
This month's program is a BASIC utility I use to design graphics made out of character size building blocks. It can also be used to design custom character sets. It includes a load and save routine so that sets can be stored on tape or, with a small change, to disk or microdrive. To use one of your custom sets in BASIC programs POKE the Spectrum variable CHARS with 256 less than the address where you load the character set (see page 173 of the Spectrum manual).

MOONLIGHT MADNESS

Bubble Bus
£7.95

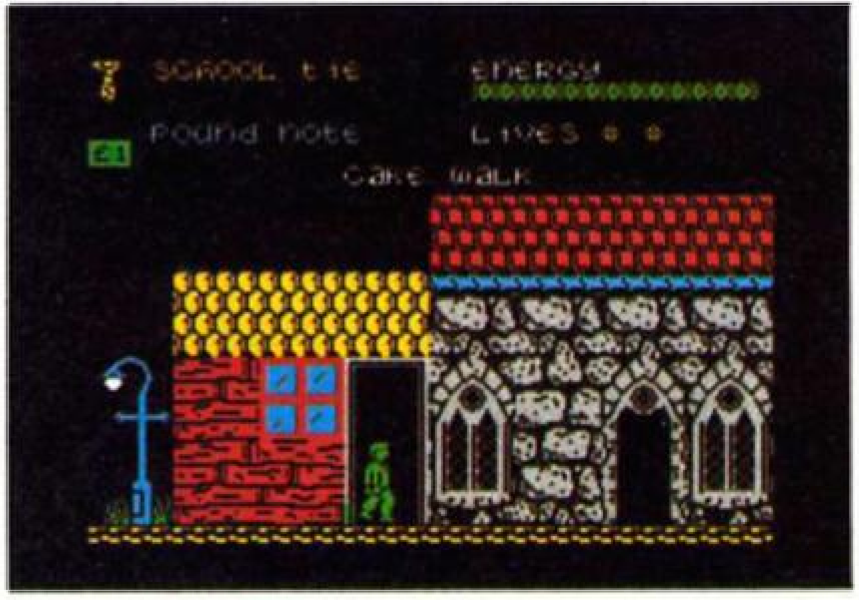
I always used to hate, loathe and utterly detest Bob-a-Job Week when I was a cub scout, and judging from the scenario of this latest Bubble Buss title, things havent changed much over the last twenty years. Having wandered down a huge drive to an old mansion, you ring the bell whereupon your call is answered by an even older man. He explains that he is a mad professor and is somewhat surprised that you have got even this far, having managed to avoid his guards and booby traps. Such is his surprise that he promptly drops down with a heart attack although not before blurting out that his pills are locked in the safe and you will need sixteen keys to open it.

Honestly, the lengths some people will go to in order to avoid paying up but you nobly decide to do your good deed for the day and try and get the pills. The mansion consists of a series of doors, switches, platforms and monsters and your task is to wander round, collecting the keys and generally avoiding getting killed. If you are standing next to a door, pressing fire will move you through into the room behind. Stand next to a switch though and the fire activates the appropriate mechanism. Not all of these are beneficial so the temptation to go round pressing everything in sight must be avoided. Typical effects include building extra platforms, activating lifts and mobilising monsters. Getting to where you want to go often requires both timing and planning. One annoying feature is that certain doors lead to a row of eight doors with a large pair of eyes above. This is some sort of maze and it is possible to get out again, it just takes some patience. So frustrating is this element of the game that two people who tried the game thought it was a bug and reloaded the game (there is no way to abort your current game).



Other features are annoying too. You can redefine the keyboard but the pause and music on/off keys are not affected so I found that every time I tried to move right, I paused the game. This is just bad programming and something that should have been picked up in playtesting. The game looks and plays very much like Booty which is no great surprise as John Cain wrote them both. Like Booty, Moonlight Madness would be all right at budget price but is not very good value as it stands. Keeping to the boy scout theme, be prepared before you buy this one.





THE HAPPIEST DAYS OF YOUR LIFE

Firebird Silver
£1.99

Firebirds latest pocket money game is a schoolboy caper featuring an animated scallywag out to retrieve the headmasters wallet that he's nicked and then lost. Can he avoid expulsion? This largely depends on your ability to steer him through the school building and grounds searching for clues to the wheareabouts of the heads dosh.

All the basics of the game work well enough, the graphics are large and bright and despite the odd annoying attribute clash, quite pleasing to look at. The number of rooms and outside locations is quite surprising and its quite a lot of fun just exploring. There are no really nasty surprises as the flying energy sappers such as hovering canes and twirling jellies in the school kitchen present little hazard. You can pick up any two objects of which there seem to be an almost unlimited supply. Quite which objects were meant to help you eluded me and all you get from the instructions are some rather cryptic hints such as, "map your way to the secret passage, turning over a new leaf en-route."

However if you like problem solving games in the hunt the object type vein this will probably appeal. Well worth the price on presentation alone it may not give you the happiest day of your life or even a wildly exciting one but if you want to settle for an afternoon of quiet diversion this game will do.



BUMP, SET, SPIKE

Mastertronic
£1.99

Here's something a little different for sports simulation fans — doubles volleyball! This latest offering from Mastertronic's Entertainment USA range is a highly enjoyable game and the first attempt to my knowledge to put volleyball on the Spectrum.

You are in control of two players against a pair controlled by the computer. Its a keyboard only game and you define your own keys. Its deceptively simple however to work out the best arrangement of keys so that you can switch easily between your two players. I had to redefine them several times before I found an arrangement that meant I wasn't accidentally hitting the wrong key. This I should add was down to my cackhandedness. To get good team work between your two players rather than just bashing the ball back first time requires a lot of dexterity on the keyboard.

Controlling the ball is quite simple, when the ball comes to your player and you are in a position to hit it you press the fire key and a cross is shown going up the court to where the ball will land. Release the fire key and the ball is on its way. It's not perhaps the most sophisticated of games but there are some nice touches — for instance if you hit the ball high into the air it crosses unexpectedly in front of the scoreboard at the top of the screen.

One game playing tip which is not down to my skill but rather an oversight in the programming is that there are blind spots from where the computer opponents seem unable to return the ball. If you serve on the left hand side of the court to a point just inside your opponents baseline they are stymied. Using this tactic will of course take a lot of the edge off the game if you choose to exploit it but its handy if you need a vital point and certainly doesn't detract from the game if you are sporting and choose to ignore the computers achilles heel.

The animation is fair and as simulations go it's pretty representative of the game itself even though its only two a side (but as even doubles will give you a bad case of entangled fingers two players are certainly enough).

For the enjoyment it gives It's more than good value for the price and seems to suggest that there is potential for someone somewhere to produce a fully fledged full price volleyball simulation. For now though Bump Set and Spike is in a class all by itself.

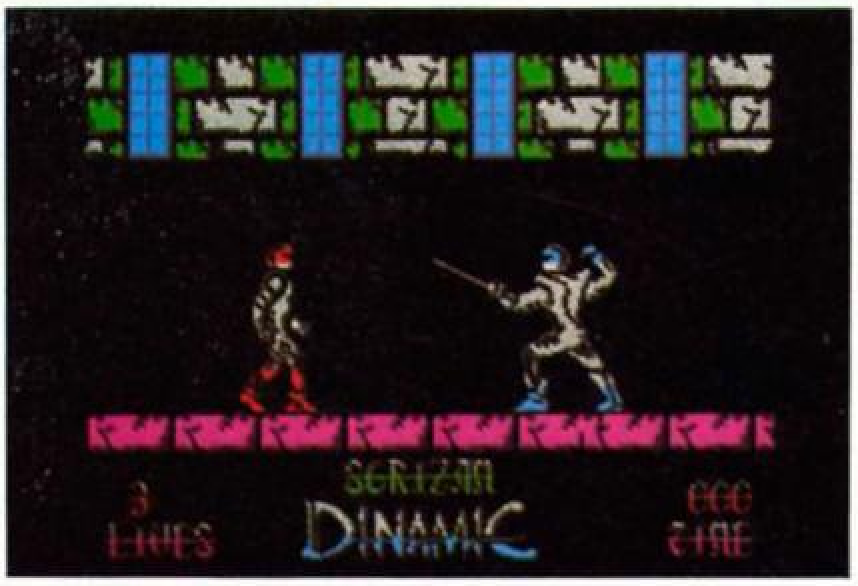
PS. There is however one curious feature about the game. The crowd, who are totally immobile throughout all seem to be wearing white masks over their faces and the entire game looks as though it's being playing in front of a gathering of the Klu Klux Klan or are my eyes deceiving me?



GAMES

SGRIZAM

Americana
£2.99



Not only does this game have an absolutely unpronouncable name, it's also virtually unplayable — or at least it's not a game that I can imagine anyone wanting to play.

Supposedly set in the 25th century, Sgrizam casts you in the role of the warrior Mirdav who has been entrusted with the task of rescuing the Princess Doxaphin. Naturally, being a princess, Doxaphin (Doxie to her friends no doubt) is dead beautiful and if you can rescue her you get to win a huge dowry and her hand in marriage, so off you go to rescue her from her captors on the planet Kindos.

Tucked away in a cell in the Castle of Kindos, Doxaphin can only be rescued by you battling your way along the corridoors, dodging or killing the creatures that will stand in your way. These include zombie ducks, swordsmen, snakes, rats and the like. Most of these can't be killed, but by ducking or jumping at the appropriate moment you can avoid them. Getting hit three times ends the game, and whenever I or anyone else played Sgrizam each game lasted about three seconds.

The trouble is that there's so little game in here — you can walk along the corridoor, jump/duck to avoid the animal, or jiggle your sword in the direction of one of the armed enemies that comes along occasionally and that's about it. There doesn't seem to be much skill involved in the swordplay and simply avoiding the stream of creatures that comes wandering along (always from the same direction, so you don't even get to run backwards and forwards to add a little variety to the game) gets monotonous in a very short time.

The graphics are quite good for a budget game, and Mirdav is a large, well animated sprite that wouldn't look out of place in a more sophisticated arcade/adventure. It's just a pity that the gameplay itself is so dreary.



RETURN TO OZ

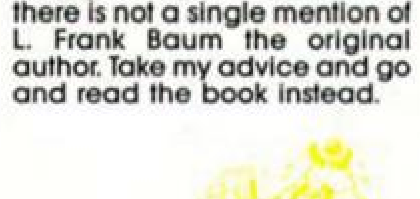
US Gold
£7.95

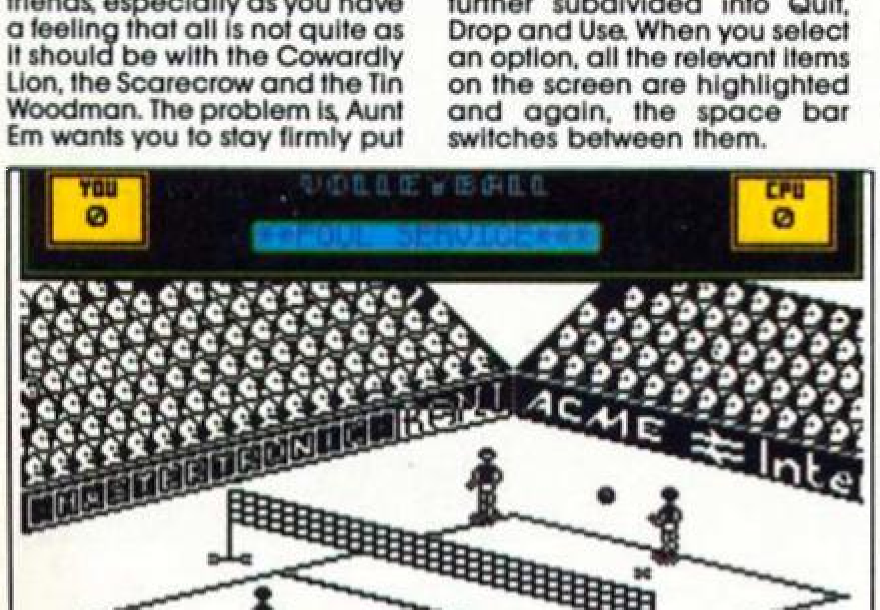
Yet another spin off from a film title and no, it's not about going back to Australia, but rather it involves that magical land of Yellow Brick Roads, Emerald Cities and the Wicked Witch of the West. Some time has passed since your first visit and you are desperate to return to see your friends, especially as you have a feeling that all is not quite as it should be with the Cowardly Lion, the Scarecrow and the Tin Woodman. The problem is, Aunt Em wants you to stay firmly put in Kansas and doesn't like you even mentioning Oz.

The game is a menu driven graphic adventure that is designed to appeal especially to young children. There is nothing to type in apart from your name at the beginning and everything is controlled by just three keys — space to highlight a particular option from the menu, enter to select that option and shift to return to the main menu. There are six main options to choose from; Look, Talk, Search, Get, List and Leave. The list option (inventory) is further subdivided into Quit, Drop and Use. When you select an option, all the relevant items on the screen are highlighted and again, the space bar switches between them.

The main problem with the game is that the graphics are truly awful. Pictures of what you have found are small and not easily distinguished and items or areas that are supposedly highlighted remain indistinct and wishy-washy. There are also difficulties in using the space bar to highlight different objects but this may be a peculiarity of my rubber keyboard. Certainly, I found that I had to press the keys up to half a dozen times before the next item was highlighted and it would be all too easy to dismiss something as not significant because you couldn't select it.

The game is designed to make adventuring easy, but I found that it had just the opposite effect and my frustration increased considerably, the longer I played. As for the story, it follows the film closely but I think that you would have to be a real fan of the film to get much fun out of the game. One final quibble. All over the packaging are notes crediting the Walt Disney organisation who produced the film but there is not a single mention of L. Frank Baum the original author. Take my advice and go and read the book instead.







THRUST

Firebird
£1.99

I'd seen this played on the C64 and I'd heard some good things about it, so I was interested to see a Spectrum version of Thrust turn up in the office a little while ago.

Thrust is a simple enough game, and quite old fashioned in many ways, being a sort of cross between Asteroids and Lunar Lander. The game puts you in control of a small space craft shaped like an upside down 'V' which hovers above the surface of a planet. At first it is slowly drifting down towards the ground, but quite quickly speeds up as gravity draws you downwards. Using the rotate left/right and thrust controls you have to guide your craft over the surface of the planet and collect the energy pod.

Controlling the craft is difficult enough, as there's gravity and your own momentum to contend with, but there's also the problem of the Lympet guns which defend the pods. The guns are powered by nuclear plants and if you can fire enough shots at the plants the guns can be deactivated temporarily, but if you overdo it you'll end up sending the plant critical and destroying the whole planet (which isn't a good idea).

Then add to all that the problem of your every decreasing fuel supply and the tricky task of collecting additional fuel from the supply pods on the surface and you're faced with a game that is made up of several simple tasks which combine to make quite a tricky game. Hovering over the planet isn't too hard, and neither is picking up the fuel or energy pods, but when you try to do both these things and conserve your fuel and dodge bullets at the same time, that's when things start to get tricky.

There is just one pod on each planet, and once you've collected that you just head skywards and the game will move you on to the next planet. The first couple of planets aren't too much trouble, but on the later planets the pods and spare fuel get hidden away in deeper and more complex caves, and are protected by more and more guns, so it's not a game you'll master in a few minutes.

The graphics are quite simple, but the animation and responsiveness of your craft are all quite smooth (and at least the simplicity of the graphics allows you to avoid attribute problems). My only doubt about the game is that because you have to go through all the planets in a fixed order you might get a bit fed up going through all those early planets in order to master the later ones. Oh, and once you've chosen your control keys you can't redefine them without starting from scratch and reloading the game again, so it's not a good idea to play the game with a friend unless he/she wants to use exactly the same controls.

Thrust is an old fashioned game that isn't going to set the world alight, but it's been well enough thought out to be challenging and fun for a few rainly afternoons and it's probably one of Firebird's better budget releases of recent months.





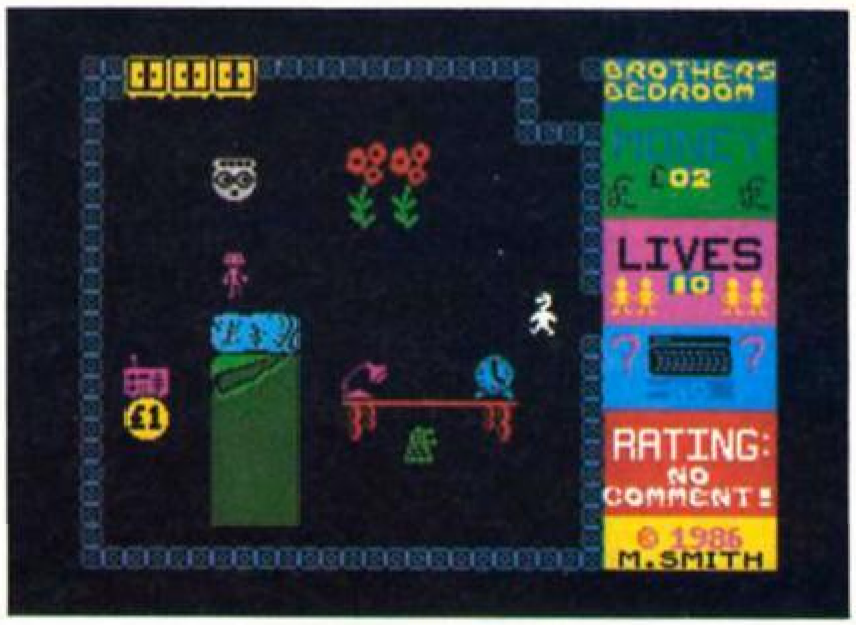
THINGY AND THE DOODAHS

Americana
£2.99

The standard of budget software seems to be varying quite a lot these days. On one hand there's a small number of games that make full price titles look overpriced, but at the same time there are still a lot of budget games that really ought never to have seen the light of day.

Then, on the other hand, there are games like Thingy and the Doodahs which are neither incredibly good nor incredibly bad and which make a poor reviewer's life hell because you can't rave over them or indulge yourself by giving them a good drubbing. So what do you say about them?

Well, the plot of the game goes like this; Thingy (a little sprite type person with an idiotic grin) has gone and broken his Spectrum and has to replace it with a new one before his parents find out. The only way for him to do this is to go of in search of the money to buy a replacement (just £60 apparently — do Americana know more abut Amstrad's plans for the Speccy than they're letting on?), and this will take him on a journey around some 200 locations, including rooms in his house and the neighbouring countryside.



As usual though, there are monsters out to get him. In this case it's a bunch of creatures known as Doodahs, which come in various types. There are Whatsisnames, Thingummybobs, Whachamacallits and So-and-So, and they're all equally deadly.

Thingy is a good old fashioned maze game with £1 coins as the objects that you've got to collect. It's not badly done, but it does look fairly dated — most of the passage ways and monsters are small character sized blocks, making use of the UDG facility, so the game looks reminiscent of others that you could have bought three years ago. It's not fast and furious, but dodging around the Doodahs and the passages of the maze is quite complex in places and the author has clearly put a bit of thought into the layout of all the rooms.

I can't really recommend Thingy and the Doodahs one way or the other. It's not such a bad game that you'll regret every penny you part with to buy it, but neither is it the sort of game that is ever likely to be remembered two months after you bought it. The word that describes it best is 'average'.

SHORT CUTS

STAR CUT

Ray Elder presents another prize winning collection of readers' routines

Printing Windows

Robert Glavas, a regular writer from Bury, sent us an unusual program to allow you to print a section or window of the screen to a ZX or Alphacom printer. The Machine code is relocatable and once you entered or reloaded the program load in your previously save screen by LOAD "" SCREEN$ and type GO TO 1. Answer the prompt and the required window will be produced. (LISTING 1)

Reflections

Mark Batts of Abbots Langly supplied us with LISTING 2, a program to drive you round the bend. Try it.

A Sound Program

Supplied by Michael Toner who lives in Belfast, this short program LISTING 5, will liven up the old Beep. Once entered and RUN just type RANDOMIZE USR 65511 to have your ears tickled.

Random Selection

Ian Skillen proves the ZX81/TS1000 is surviving in Chicago, but his routine for selecting non repeated random numbers over a set number of selection can be used on any ZX computer. (LISTING 6)

QL Dump

A rare beast indeed from Mr L.W. Tomlin of London. More of a tip really but for slowly fading QL. He tells us that there is a screen dump utility on the EASEL cartridge called GPRINT_PRT and should work with any Epsom type printer.

Load it by typing a=RESPR(2000):LBYTES MDV1_GPRINT_PRT,a

When you wnt to print a screen either from within a progra or after loading a predesigned screen simply type CALL a

I tried it with a Shinwa CP80 and it worked.

Break & Display

From the Netherlands comes LISTING 3 and 4, courtesy of Rene Uittenbogaard, one is to add some protection to your. program (3), and (4) splits the screen into two sections, the top half staying unchanged but the lower half scrolling up. Great for adventure game writers.

Your program goes from line 20 onwards and all you have to do is use the command PRINT FN p$("Text goes in the string") with a maximum of 32 characters at a time.

LISTING 3

```
  5 CLEAR 60898: FOR f=60899 TO
 60970: READ a: POKE f,a: NEXT f
 10 DATA 205,124,0,59,59,225,1,
15,0,9,235,42,61,92,115,35,114,2
01,118,205,142,2,123,254,255,32,
248,58,58,92,254,12,40,10,254,16
,40,6,254,20,40,2,24,25,60,50,12
9,92,253,54,0,255,33,0,0,34,66,9
2,33,0,0,34,68,92,59,59,195,125,
27,195,3,19
 15 REM initialise with
         RANDOMIZE USR 60899
```

LISTING 4

```
  5 CLEAR 59999: FOR f=60000 TO
 60081: READ a: POKE f,a: NEXT f
 10 DATA 33,0,72,205,141,234,33
,224,72,22,8,30,32,1,32,7,9,126,
1,224,248,9,119,35,29,123,179,32
,240,1,224,0,9,21,122,178,32,229
,33,0,80,205,141,234,201
 11 DATA 22,8,30,224,1,32,0,9,1
26,1,224,255,9,119,35,29,123,179
,32,240,1,32,0,54,0,35,11,120,17
7,32,248,21,122,178,32,222,201
 15 DEF FN p$(a$)=(STR$ USR 600
00)( TO 0)+CHR$ 22+CHR$ 21+CHR$
0+a$
```



LISTING 1

```
 10 LET MC=60000
 20 RESTORE : FOR F=MC TO MC+54
 READ A: POKE F,A: NEXT F
 30 DATA 62,3,205,1,22,1,0,0,81
,122,254,0,40,+6,62,32,215,21,24
,-11,42,132,92,9,197,17,248,255
 40 DATA 6,8,126,18,197,1,0,1,9
,193,19,16,-11,62,164,215,193,3,
121,254,32,32,-31,62,13,215,201

 90 REM general program
 100 INPUT "STARTING ROW?";R1: I
F R1<0 OR R1>21 THEN GO TO 100
 110 INPUT "FINISHING ROW?";R2:
IF R2<R1 OR R2>21 THEN GO TO 110
 120 INPUT "STARTING COLUMN?";C1
: IF C1<0 OR C1>31 THEN GO TO 12
0
 130 INPUT "FINISHING COLUMN?";C
2: IF C2<C1 OR C2>31 THEN GO TO
130
 140 POKE MC+6,C1: POKE MC+48,(C
2+1)
 150 FOR F=R1 TO R2: PRINT AT F,
0;: LET L=USR MC: NEXT F
```

LISTING 2

```
 10 LET x=0: LET y=0
 20 INPUT "Type a word in pleas
e",a$
 30 IF LEN a$>10 THEN GO TO 20
 40 PRINT AT 0,0;a$
 50 FOR s=0 TO LEN a$+8
 60 FOR t=175 TO 168 STEP -1
 70 IF POINT (s,t)=1 THEN GO SU
B 105
 80 LET y=y+5: NEXT t: LET y=0:
 LET x=x+3: NEXT s
 100 STOP
 105 PLOT 0,75: DRAW 255,0
 110 PLOT x,75+(y-3): DRAW 0,3
 120 PLOT x,75+y: DRAW 3,0
 130 PLOT x+3,75+y: DRAW 0,-3
 140 PLOT x+3,75+(y-3): DRAW -3,
0
 150 CIRCLE x+1,80-(y+3),2
 160 RETURN
```

LISTING 5

```
  10 LET A=USR "P"
  20 FOR J=0 TO 31: READ B: POKE
 A+J,B: NEXT J
  30 DATA 6,255,197,6,75,62,6,21
1,254,16,252,193,16,236,201
  40 DATA 33,0,10,43,126,211,254
,6,205,5,32,-3,175,132,200,24,-1
4
```

LISTING 6

```
  10 PRINT "HOW MANY SELECTIONS
?";
  15 INPUT NS
  16 PRINT NS
  20 PRINT "HOW MANY GAMES ?";
  22 INPUT NG
  25 PRINT NG
  30 DIM N(NG)
  40 FOR I=1 TO NS
  50 LET K=INT (RND*NG)+1
  60 IF N(K)<>0 THEN GOTO 50
  70 LET N(K)=K
  90 NEXT I
  92 STOP
 100 PRINT "NUMBERS"
1000 FOR I=1 TO NG
1010 IF N(I)<>0 THEN PRINT N(I)
1020 NEXT I
1030 COPY
1040 STOP
```

Program Protection 2

Tony Porritt offers the system they use in Cleveland for protecting their listings. At this rate ZXC readers will have the most secure programs in the world: Remember to only add line one after your program has been tested and is bug free else you're stuck...... LISTING 7

LISTING 7

```
   1 LET L=USR "a"-768: POKE 236
06,L-256*INT (L/256): POKE 23607
,INT (L/256)-1: DATA 15,30,12,1,
9,31
   2 REM program starts here
```

3D Window & Text

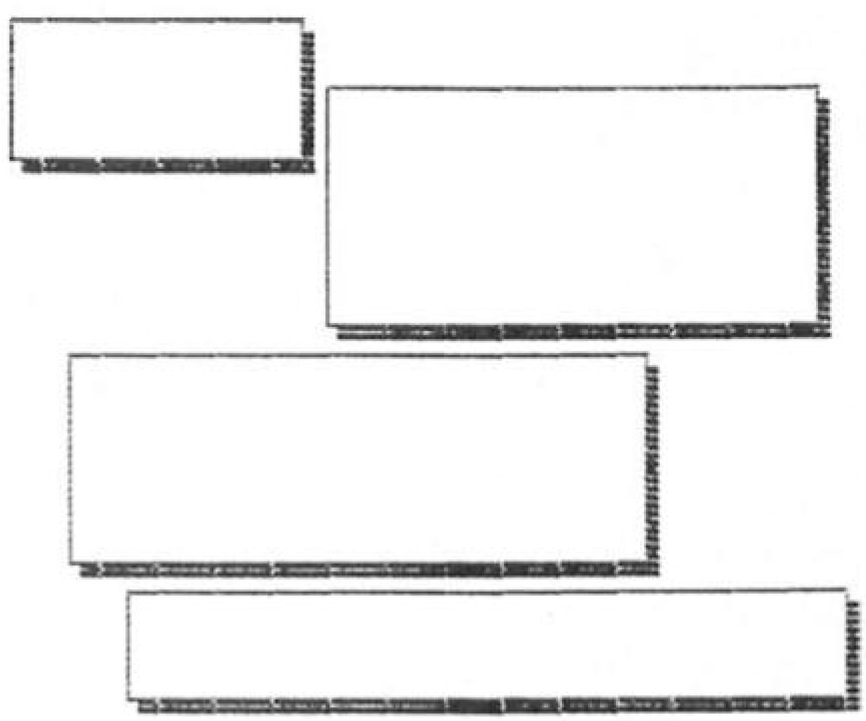
Two from Alec Goodyear, the first to draw a 3D 'window' on the screen, as written you have to enter the positions, but these dimensions could be built into a program, and the second to allows text to be input to a specified position and length. Again the parameters could be built into a program if desired.

LISTING 8

```
  10 REM 3D Window
  20 INPUT "ROW : ";R
  30 INPUT "COLUMN : ";C
  40 INPUT "HEIGHT : ";H
  50 INPUT "WIDTH : ";W
  60 INPUT "COLOUR : ";P
 100 LET A$="
            "
 110 FOR N=R TO R+H-1
 120 PRINT AT N,C; PAPER P;A$(1
TO W)
 130 NEXT N
 140 LET R=(22-R)*8
 150 LET C=C*8-1
 160 LET H=H*8+1
 170 LET W=W*8+1
 180 PLOT C,R: DRAW W,0: DRAW 0,
-H: DRAW -W,0: DRAW 0,H
 190 FOR N=R-H-1 TO R-H-3 STEP -
1
 200 PLOT C+3,N: DRAW W,0
 210 NEXT N
 220 FOR N=C+W+1 TO C+W+3
 230 PLOT N,R-3: DRAW 0,-H
 240 NEXT N
```

LISTING 9

```
  40 INPUT "ROW : ";R
  50 INPUT "COLUMN : ";C
  60 INPUT "MAX NUMBER OF CHARS
: ";M
  70 PAUSE 50
 100 LET I$="": LET A=0: LET B=0
 110 PRINT AT R,C;
 120 IF B=50 THEN LET B=0
 130 LET B=B+1
 140 IF B<26 THEN PRINT "_";: GO
TO 160
 150 PRINT " ";
 160 PRINT CHR$ 8;
 170 LET K$=INKEY$: IF K$="" THE
N GO TO 120
 175 PAUSE 70
 180 LET K=CODE K$
 190 IF K<32 OR K>127 THEN GO TO
220
 200 IF A=M THEN GO TO 120
 210 PRINT K$;: LET A=A+1: LET I
$=I$+K$
 220 IF K<>12 THEN GO TO 250
 230 IF A=0 THEN GO TO 120
 240 PRINT CHR$ 8;" ";CHR$ 8;CH
R$ 8;: LET A=A-1: LET I$=I$(1 TO
 A)
 250 IF K<>13 THEN GO TO 120
 260 PRINT " "
 300 PRINT AT 21,0;"THE STRING W
AS : ";I$
```



Topsy Turvey

This routine is an interesting means of turning the display upside down, the code is loaded to 32000 but it is relocatable and can be placed anywhere you wish. Use it by RANDOMIZE USR address. This one came from the I.O.W. and was written by David Hiscock. LISTING 10.

LISTING 10

```
  10 RESTORE : LET t=0
  20 FOR l=0 TO 117: READ a: POK
E 32000+l,a: LET t=t+a: NEXT l
  30 IF t<>10795 THEN PRINT "ERR
OR IN DATA!!"
  40 DATA 17,0,88,33,160,90,6,11
,14,32,197,26,79,126,18,113,193,
19,35,13,32,244,197,1,64,0
  50 DATA 167,237,66,193,16,232,
17,0,64,33,160,87,6,6,14,3,197,6
,8,197,6,32,26,79,126,18,113
  60 DATA 19,35,16,247,229,235,1
,224,0,9,235,225,1,32,1,167,237,
66,193,16,227,229,235,1
  70 DATA 224,7,167,237,66,235,2
25,9,193,16,210,13,121,254,2,32,
10,17,192,64,33,224,79,6,2,24
  80 DATA 194,254,1,121,192,17,0
,72,33,160,79,6,3,24,180
```

SHORT CUTS

N:E:X:O:R

The Andromedans (alien flavour of the month) have been tinkering with an awesome doomsday machine. Your interstellar DIY mission is to put it back together.

NEXOR
Design Design
£7.95

They're at it again — it seems that you can't go near a computer game these days without being beseiged by hordes of aliens who are just falling over themselves to invade earth and sample the delights of Miami Vice, Stephen King novels and Chinese take aways.

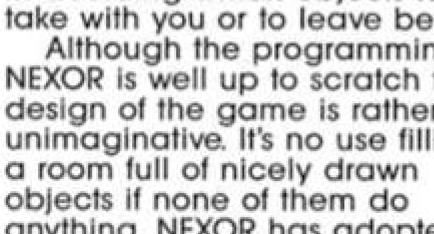
This time it's the Andromedans' turn to have a crack at us, but fortunately Our Boys have been eating their Shredded Wheat and have pushed the Andromedans back to their last stronghold in Orion. All it needed was the completion of the Nemisis weapon to knock out that last stronghold, but then the enemy launched a desperate attack on the NEXOR complex where the weapon was being developed and only your have survived to keep Nemisis out of their hands. Scattered throughout the multi-levelled complex are two sets of five modules; only one set needs to be collected to complete the weapon (the others will be destroyed when the complex self-destructs), then if you can find the blueprints and repair the transporter beam you'll be able to keep the weapon out of enemy hands.

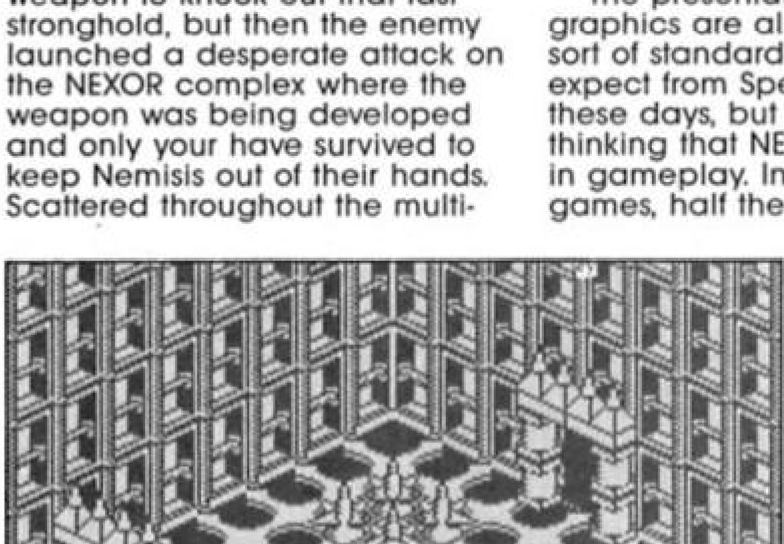
The NEXOR complex might look a little familiar, since it's drawn in the style of two-coloured 3D grapics that have become more or less standard for arcade adventures ever since Knight Lore arrived to harry the nation's joysticks. All the usual paraphenalia is here — moving walkways, deadly spikey things, tables and blocks that have to be moved around to allow you to reach inaccessible exits, and an assortment of robot sprites who sneakily don't appear until the second time you pass through a room so that they can take you by surprise.

You control the figure of the head of secruity as he attempts to locate the Nemesis modules and repair the transporter beam. You're unarmed so the only way to avoid fatal collisions is by some smart movements using the usual back/forwards, left/right and jump controls. Modules are collected automatically as soon as you touch them, and as you collect each one you are told how long is left before the whole place self-destructs.

The presentation and graphics are all well up to the sort of standard that people expect from Spectrum games these days, but I couldn't help thinking that NEXOR lacked a bit in gameplay. In these sort of games, half the fun lies in the exploration and discovery of all those fiendish obstacles that lie between you and the objects you need to collect, and working out how to get past them. But in Nexor much of the exploration simply isn't very interesting, and finding the modules just becomes a matter of wandering around for long enough. Many of the rooms are full of bombs and other features, yet present no other challenge than walking in and then straight out again because the path through the room is totally obvious. There are a few rooms with some tricky puzzles in, but these are in the minority. Apart from the modules you can only carry one object at a time, and most of these can only be used in the room where you found them (you can't carry them from room to room), so there's no element of judgement involved in deciding which objects to take with you or to leave behind.

Although the programming of NEXOR is well up to scratch the design of the game is rather unimaginative. It's no use filling a room full of nicely drawn objects if none of them do anything. NEXOR has adopted the style of gmes like Knight Lore and Alien 8, but with little of the substance that made them so addictive.







DAN DARE
PILOT OF THE FUTURE

Dan and the Mekon are back! Dan Dare takes to the skies in the new biggie from Virgin.

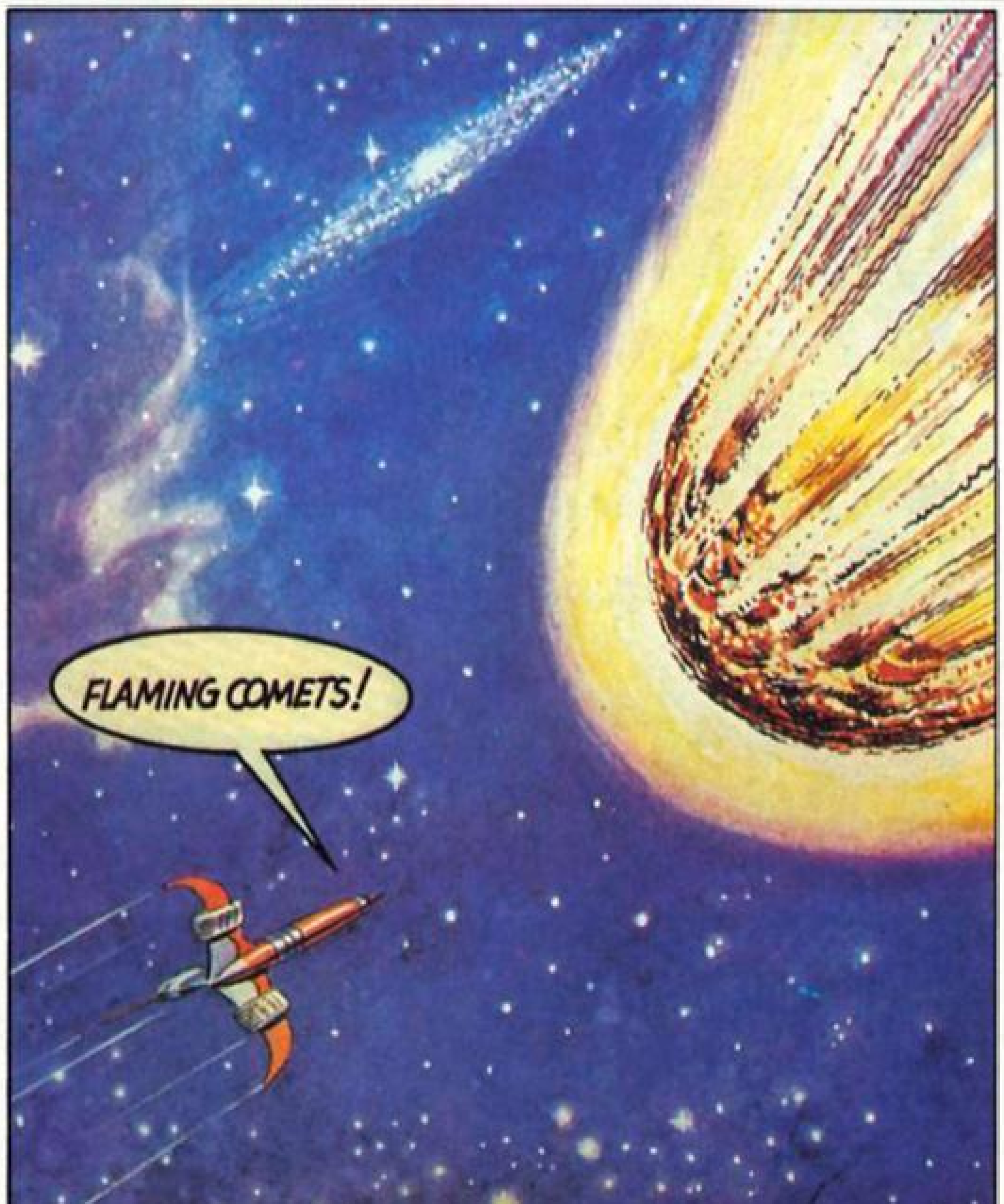
Dan Dare
Virgin Games
£9.95

It didn't take Virgin quite as long to get Dan Dare off the ground as it's taken Ocean with Knight Rider, and now that the game is here it seems that, just for once, the wait was worthwhile. In fact, Dan Dare bears some resemblance to another Ocean game in that it's like a simpler version of 'V'. That's not necessarily a criticism since although 'V' was quite impressive I found that game a bit too fiddly to be addictive.

Dan Dare, on the other hand, does away with the complicated icons of 'V' and concentrates on shoot'em-up action, with just a touch of platform hopping and object collecting thrown in as well.

The plot is explained by a small comic strip that comes packaged with the game: it all starts when a futuristic Eamonn Andrews jumps and tells Dan — 'This is your life'. At which point none other than the Mekon himself makes a comeback and takes over the broadcast, warning all the television viewers in the Solar System of his latest fiendish plot. Heading towards the Earth under Mekon's control is a hollowed out asteroid large enough to destroy the entire plant upon collision. Inside the asteroid is a self-destruct mechanism that can be activated in order to prevent the collision, but the Mekon will only destroy it if he is declared absolute Ruler of The Universe.

Of course it goes without saying that having old green bonce as ruler of the universe is a prospect too horrifying to contemplate, so the Earth authorities have no choice but to refuse his demands. But how to save Earth?



By permission of IPC Magazine Youth Group

Dan's jaw

Back in the television studio Dan's jaw sets firmly, his upper lip goes thrusting skywards and Digby (the ever faithful, but utterly useless sidekick) heads for the nearest McDonalds, prior to getting Dan's ship, The Anastasia, ready for take-off.

Within minutes Dan and Digby are on course for the asteroid, ready to take on the Mekon and his followers, the Treens, and save the Earth — just like the good old days.

Inside, the hollowed out asteroid contains five sectors full of armed Treens, corridors, ledges and grav shafts (lifts, connecting different floors). In each of the five sectors there is one control element for the self destruct mechanism, and only when you've collected the element in each sector can you gain entrance to the next sector





(assuming that you've managed to find the relevant locked door).

The screen display shows two floors at a time, so you can often see the Treens descending from above or an object that you might want to collect, and this adds to the atmosphere of the game as it makes the action seem that much busier when you can see other things going on around Dan.

You've got just two hours to complete your mission, and although Dan can't be killed, the Treens can shoot at him and drain his energy which results in Dan being captured and carried off to the cell in that sector. Escape from the cells is no problem but it does result in the loss of ten minutes and much of the skill of the game lies in how good you can get at fighting the Treens as the loss of time is likely to be your major obstacle in completing the game.

The laser that Dan is armed with isn't 100% accurate, and your supply of ammunition is limited so you have to learn when to stand to fight and when to turn that jaw in the other direction and run away, hopefully to pick off the Treens on some other screen.

The action here is similar to that in the Commando type of game, with Dan and his laser faced with hordes of alien soldiers, but the animation is unusually good and the element of uncertainty in using your laser adds to the excitement of the game. And, since getting shot too many times cuts into your time rather than simply killing you, combat becomes a matter of judgement instead of just endless rounds of kill or be killed. As I mentioned earlier, there are times when running away is the wiser course.

Inside the asteroid

The five sectors of the asteroid add up to quite a lot of corridors, and it's probably a good idea to make some sort of rough map as you go along in order to remember some of the important locations. Each time you find one of the control elements these have to be returned to the control room in sector 1. That's my only doubt about the game, since you have to go through the sectors in a fixed order and return to the control room each time you might eventually get a bit bored with retracing the same routes by the time you get into the final couple of sectors. Of course the system of corridors is sufficiently complex to allow you to vary your routes but I found it quicker to get around, and easier to remember my way, if I stuck to roughly the same routes each time.



Still, that doesn't make Dan Dare any less enjoyable especially as the main challenge is to improve your combat skill and speed in completing the mission, not simply finding the correct route through the corridors. And the game is very well presented, with one nice touch being the way that the screen is presented like a frame from a comic strip, with captions and messages from the Mekon flashing on screen every now and then in the same way that dialogue accompanies the illustrations in a comic.

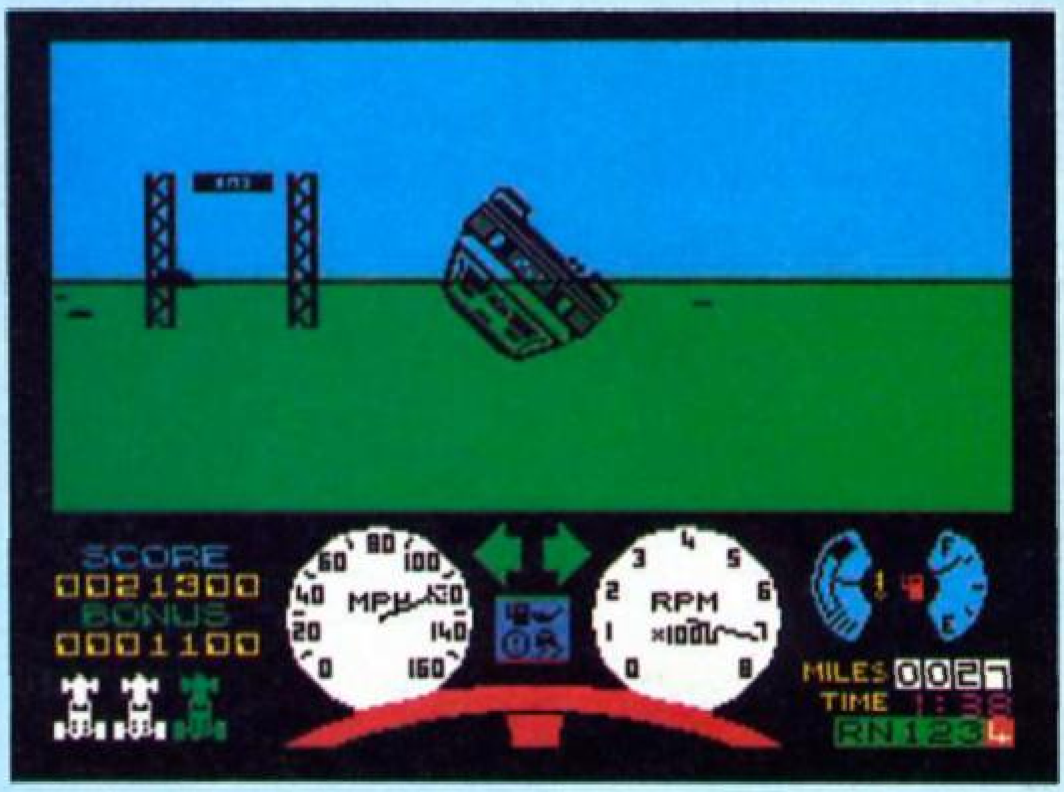
Because of the delay in getting the game ready, and the speed at which software is improving these days, Dan Dare isn't quite as impressive as it might have been had it appeared last year when it was originally planned, but even so it's still a highly enjoyable and playable game and might even be the best game yet from Virgin.



NIGHTMARE RALLY

A top notch rally game from Ocean with detours into other dimensions.

Nightmare Rally
Ocean
£7.95



This rally is far more than a straightforward dash from A to B over difficult terrain. Strange effects have been built into the route which transforms it into a trip through the Twilight Zone.

Not only do you have to complete each stage within a set time limit you must also amass bonus points to progress and to do this you get involved in some very strange manoeuvres indeed.

The landmarks dotted over the landscape include marker flags indicating the course and fuel dumps which give you an instant tank up. It's important to go round the flags the right side as indicated on the dashboard layout below the main screen and refuelling is essential for you to reach the finishing gate on the first stage.

That's the basic set up but there are many surprises in store. Reckless driving is in fact encouraged — make for the hills, jump over them and you get bonus points and if you hit them at an angle the car does an impressive somersault and you get even more points.

But if you want to sample the more bizarre side of the rally head for the magical standing stones. Ramming these head on produces a strange range of effects. Sometimes you will just stop dead but particular stones will variously cause your car to take to the air, reverse the screen so you are driving with the sky above you or create a curious blurring effect on the landscape.

All these effects are graphically so convincing thai you immediately accept them as a part of the game. The perspective shifts are also very smooth giving a real sensation of speed. Searching out all the screen effects can distract you from reaching your objective but it's worth expending a few lives just to test them out and after a few outings you discover that they can actually help you. Flying over trees and obelisks, for instance, brings in some valuable bonus points. On some stages, diversions from the main course are essential to collect the necessary bonus points.

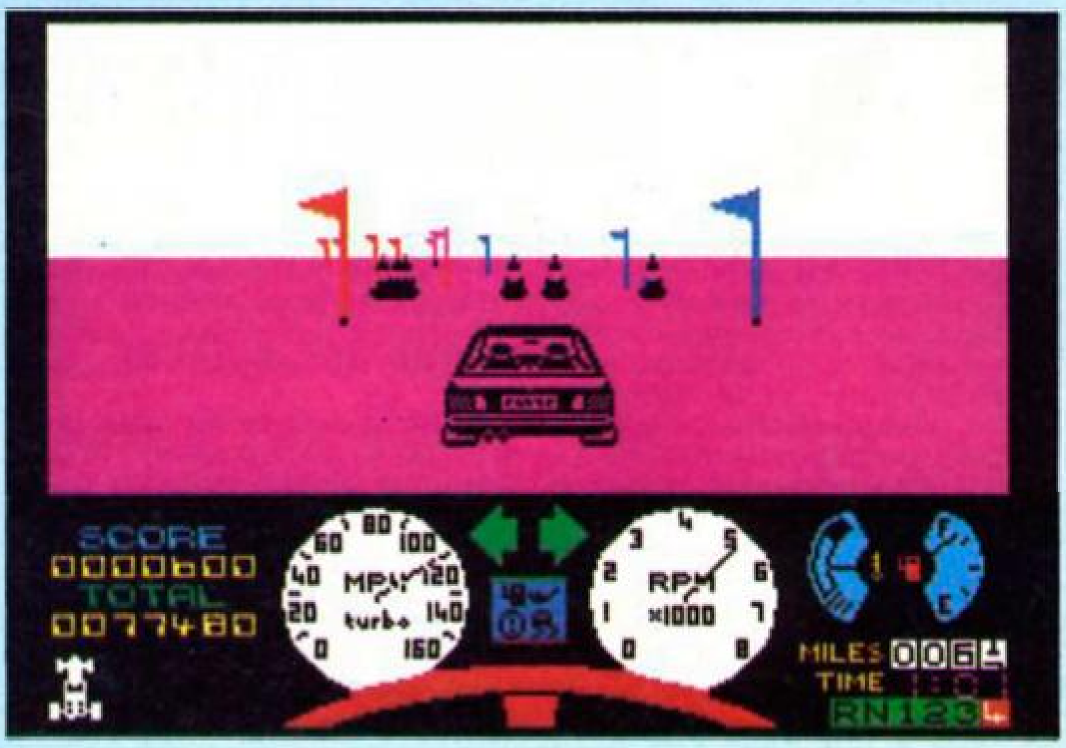
Sometimes the weather conditions alter and fog descends cutting visibility down completely so that objects seem to appear out of nowhere. Each stage presents different challenges.

The first is a flat plain strewn with objects to collide with for a bonus point accumulation and others to strictly avoid. Only experience will tell you which is which.

The second stage is completely different being a winding course over an icy surface. Choosing the correct speed is important here but you'll find that even successfully getting through the flag gates is not enough to get you to the next stage. Also on the course are numerous cones which you must clip to get you the required points tally.

Nightmare Rally is an excellent game for anyone who likes being behind the wheel of a computer simulation. Very challenging, very unpredictable, this game is highly addictive and highly recommended.





SPECTRUM EXTEND

Interface One owners can add some new commands to Sinclair Basic, with this program.

By Paul Matthews

Any Spectrum user with Interface 1 will know that it allows the use of microdrives, RS232 devices, and the ZX net. However, a less well known feature is the facility to extend Spectrum BASIC by changing the syntax of certain keywords. This feature is used by my routine to provide the Spectrum user with six extra commands. They are:—

PLOT *x,y which will PLOT anywhere on the screen. Co-ordinates (O,O) are in the very bottom left hand corner, where it is not usually possible to PLOT. So the normal PLOT O,O becomes PLOT *O,16 and PLOT O,175 becomes PLOT *O,191.

POKE *a,b where b is a number from 0-65535. This means that a 16-bit number can be POKEd with the low order byte going into location a and the high order byte going into the location (a+1), i.e. POKE *23735, 64900 is equivalent to POKE *23735, 132 and POKE 23736, 253.

**POKE AT a, a$** which will POKE a hexadecimal value held in a$ into location a. e.g. POKE AT 23296, "C9" will put C9 (201 dec) into location 23296. If there is more than one hexadecimal number (which can be separated by spaces) then they are put into the locations following a. e.g. POKE AT 23296, "3EFEC9" and POKE AT 23296, "3E FE C9" are both possible and will POKE 3E (62 dec) into location 23296, FE (254) into 23297 and C9 (201) into 23298. Therefore, this command could be used to enter machine-code easily and quickly.

ATTR f,p,i TO nf,np,ni which will search out specified attributes and replace part or all of each found, with a new attribute. f is the flash and brightness intensities to be changed (0=none, 1=bright only, 2=flash only, 3=both, 8=any combination). p and i are the PAPER and INK colours to be changed (07=Black-White, 8=all colours). nf is the new flash and brightness intensities (0=neither, 1=bright, 2=flash, 3=both, 8=leave as they are). Finally, np and ni are the new PAPER and INK colours (0-7=Black-White, 8=leave as it is, 9=contrast).

ATTR 1,0,6 TO 1,7,1 will change all characters which are bright yellow INK on black PAPER to bright blue INK on white PAPER.

ATTR 8,8,0 TO 1,8,9 will change all characters with black INK (no matter the PAPER colour and flash or brightness intensities) to bright contrast INK (either black or white) on the original PAPER colour. Note: one of the powerful features of this command is that the whole screen can be changed by setting the first three numbers to '8'. e.g. ATTR 8,8,8 TO 8,1,8 will change the entire screen to blue PAPER.

INVERSE which swops over the PAPER and INK colours in the attribute file, without altering the flash and brightness intensities.

SIN which stands for "Screen INvert" and will invert all the characters on the screen.

NOTE: INVERSE and SIN have the same visual effect but INVERSE affects ATTR (as bits 0,1 & 2 and bits 3,4 & 5 of each attribute are swopped over) but does not affect POINT whilst SIN affects POINT (as all pixels set are reset and vice-versa) but not ATTR.

New Vector

The key to extending the BASIC is the "new" system variable VECTOR. Whenever a syntax error occurs, the "shadow" ROM (i.e. that in Interface 1) is paged in and checks the syntax to see if it's one of the Micro-drive commands, if not, it jumps to the



ED BASIC

location pointed to by VECTOR. Therefore, to add commands you need only change VECTOR to point to the beginning of your own machine-code routine. This is done in the set-up routine (lines 210 - 240 of the assembly listing. Lines 210 - 20 set up the "new" system variables, as these are not usually set-up until an error occurs). Before you can use the extra commands the set-up routine must be called by a line such as LET a = USR 64900 (64900 is where I placed the routine in memory. However, if you assemble it, you can put it anywhere you want by changing Lines 10 and 20. Once assembled, though, the routine cannot be moved.

When a non standard command is found, the routine will be called. However, the "shadow" ROM will be paged in which means that if you want to use of any of the "main" ROM routines you cannot use call as that would CALL a "shadow" routine. Instead, you have to use an RST # 0010 instruction (#denotes hex) followed by the address of the "main" routine. This "shadow" routine pages in the "main" ROM, calls the required "main" routine and then pages out the "main" ROM. In this article, the restart routine has been given the name CALBAS. On entry to the routine pointed to by VECTOR, another system variable CH-ADD will be pointing to the command of the error-statement. To find out what command this is, the "main" routine GET CHAR (#0018) is used. This puts the character pointed to by CH-ADD into the A register. The next characters in the statement can then be found by using NEXT CHAR (#0020) which increases CH-ADD by one and puts the character now pointed to, into the A register.

Keywords

I have used five different keywords and, as you can see, they do not all have to be common commands e.g. ATTR (a function) is being used. Also note that if the syntax of your new command is the same as that of the original (e.g. PLOT and POKE in this article) another character (e.g. an asterisk) has to be placed after the keyword or the ROM will not recognise it. However, if you do not like the present keywords, you can make your own up e.g. *Renum or *Delete, but these have to be typed out in full and each letter must be checked by increasing CH-ADD and then comparing the character. The asterisk, which could be any shifted symbol, is necessary to get the cursor out of K mode.

It can be seen (in lines 270 - 380) that the first thing to be done, once called by the ROM, is to get the code of the error command into A, and then check this against the five new commands (not six, POKE is used twice). If one of these commands is found the appropriate routine is called, otherwise the normal error routine (#01F0) is called instead, which produces either a flashing question mark or 'nonsense in BASIC'.

Interesting points to note in my routine include the fact that POKE is used twice. This is done by jumping to the second routine when the asterisk after the POKE fails. Only if the character also fails to be an "AT" sign is the error routine called. Another thing to be noted is that the ATTR scanning routine (i.e. when it actually looks at the screen) is really only four lines in length. However, this is increased by also allowing ATTR to have new INK and PAPER colours 9 which cannot be easily changed in the same way.

I hope my routine provides useful commands and that this article will help you to write your own commands. For further information, I advise you get Ian Logan's "Spectrum Microdrive Book", and "The Complete Spectrum ROM Dissembly" by Ian Hogan and Frank O'Hara.

Listing 1

Machine code loader.

```
   1 REM  Six Extra Commands
            by Paul Matthews
  1Ø CLEAR PEEK 23732+256*PEEK 2
3733-636: LET p=PEEK 2373Ø+256*P
EEK 23731+1: LET m=253-PEEK 2373
1
  2Ø CLS #: PRINT TAB 3; FLASH 1
;" Loading ~6 Commands~ Data "''
  3Ø FOR f=p TO p+439 STEP 2Ø: L
ET t=Ø: FOR g=Ø TO 19
  4Ø READ a: LET t=t+a: POKE f+g
,a: NEXT g
  5Ø READ s: IF s<>t THEN  PRINT
 "Error in Data Between Addresse
s"'f;" and ";f+19: BEEP .1,Ø: BE
EP .1,12: BEEP .1,8: STOP
  6Ø NEXT f: PRINT TAB 5;"Data E
ntered Correctly": BEEP .1,12: B
EEP .1,15: BEEP .1,2Ø: BEEP .1,1
5
  7Ø PRINT ''''The Routine will
now be saved."''''To re-load the
 routine at any   time, enter th
e following:''''CLEAR ";p-1';"LO
AD """"CODE ";p;",44Ø"''"LET a=US
R ";p
  8Ø SAVE "6 Commands"CODE p,44Ø
  85 REM or to Microdrive with:
 SAVE *"m";1;"6 Commands"CODE p,
44Ø
  9Ø PRINT ''''To use the Extra
Commands now,  you need only ent
er the last of the above stateme
nts."
  95 REM Address 649ØØ:
 1ØØ DATA 2Ø7,49,33,141,253-m,34
,183,92,2Ø1,215,24,Ø,254,246,2Ø2
,189,253-m,254,244,2Ø2,3276-m-m
 11Ø DATA 225,253-m,254,171,2Ø2,
122,254-m,254,221,2Ø2,84,254-m,2
54,178,194,24Ø,1,215,32,Ø,361Ø-(
m*3)
 12Ø DATA 2Ø5,183,5,33,Ø,64,126,
47,119,35,124,254,88,32,247,24,3
3,215,32,Ø,1866
 13Ø DATA 254,42,32,43,215,121,2
8,2Ø5,183,5,215,148,3Ø,245,215,1
48,3Ø,193,79,62,2493
 14Ø DATA 191,144,56,96,215,176,
34,215,236,34,195,193,5,215,32,Ø
,254,172,32,83,2578
 145 REM Address 65ØØØ:
 15Ø DATA 215,128,28,254,44,4Ø,2
,231,Ø,215,32,Ø,215,14Ø,28,2Ø5,1
83,5,215,241,2421
 16Ø DATA 43,197,213,215,153,3Ø,
197,225,2Ø9,193,12Ø,254,Ø,32,7,1
77,254,1,4Ø,223,2783
 17Ø DATA 56,2Ø4,26,254,32,4Ø,19
,2Ø5,46,254-m,7,7,7,7,11,197,79,
19,26,2Ø5,17Ø1-m
 18Ø DATA 46,254-m,177,193,119,3
5,19,11,24,216,214,48,254,1Ø,216
,214,7,254,16,216,2543-m
 19Ø DATA 215,249,36,254,42,32,1
76,215,121,28,2Ø5,183,5,215,153,
3Ø,197,215,153,3Ø,2754
 195 REM Address 651ØØ:
 2ØØ DATA 197,225,193,113,35,112
,24,138,215,32,Ø,2Ø5,183,5,33,Ø,
88,126,23Ø,192,2346
 21Ø DATA 87,126,7,7,7,23Ø,56,17
8,87,126,15,15,15,23Ø,7,178,119,
35,124,254,19Ø3
 22Ø DATA 91,32,23Ø,195,193,5,21
5,121,28,215,125,28,254,2Ø4,194,
239,253-m,215,121,28,2986-m
 23Ø DATA 215,125,28,2Ø5,183,5,2
15,148,3Ø,254,1Ø,48,55,254,8,56,
7,1,7,7,1861
 24Ø DATA 32,2,14,Ø,197,215,148,
3Ø,254,1Ø,48,36,254,8,56,9,1,56,
56,32,1458
 245 REM Address 652ØØ:
 25Ø DATA 8,14,Ø,24,4,7,7,7,79,2
25,9,229,193,2Ø5,37,255-m,197,21
5,148,3Ø,1893-m
 26Ø DATA 6,255,254,8,4Ø,5,56,6,
215,68,34,1,Ø,248,197,215,148,3Ø
,193,254,2233
 27Ø DATA 8,32,6,12Ø,238,56,71,2
4,7,48,233,7,7,7,177,79,2Ø5,37,2
55-m,2Ø9,1826-m
 28Ø DATA 33,Ø,88,126,16Ø,185,32
,4Ø,126,162,171,119,123,162,2Ø3,
87,4Ø,18,23Ø,56,2161
 29Ø DATA 126,4Ø,2,238,56,246,7,
2Ø3,111,4Ø,2,238,7,119,123,162,2
3Ø,56,4Ø,8,2Ø54
 295 REM Address 653ØØ:
 3ØØ DATA 182,2Ø3,87,4Ø,2,23Ø,19
9,119,35,124,254,91,32,2Ø5,195,1
93,5,197,215,148,2756
 31Ø DATA 3Ø,193,254,8,32,5,12Ø,
238,192,71,2Ø1,254,4,48,149,15,1
5,177,79,2Ø1,2286
```

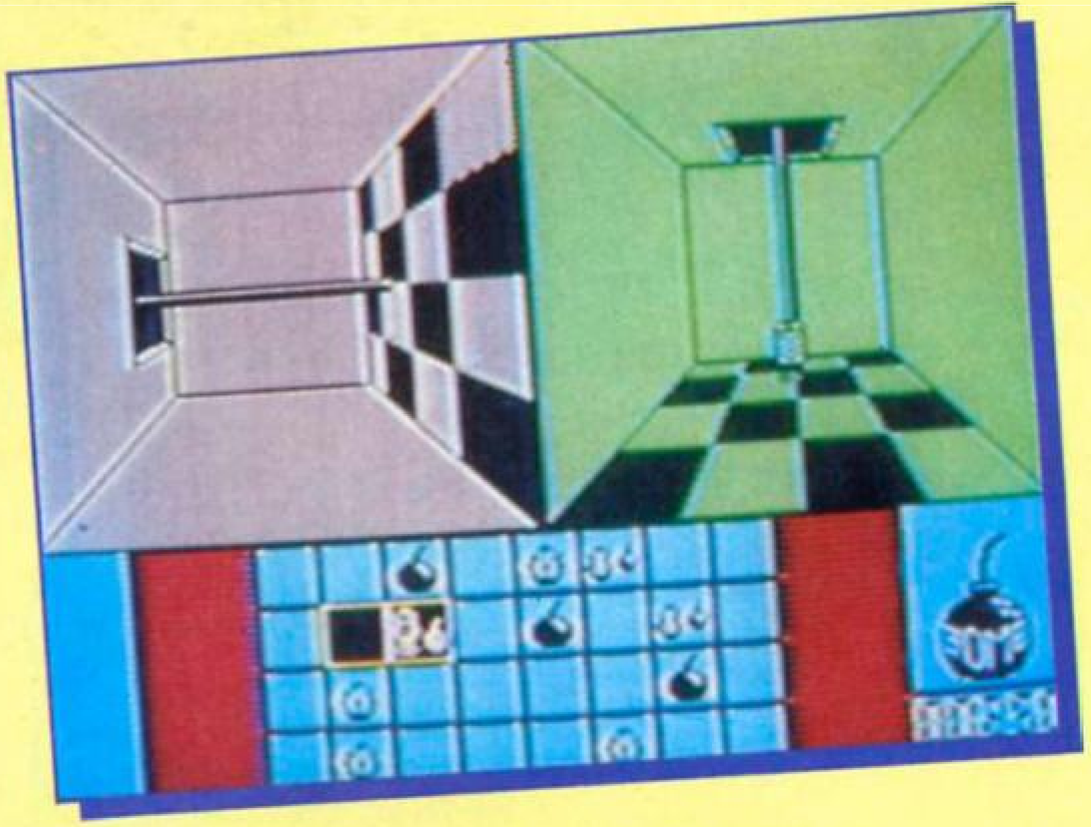
Listing 2

Demo of the new commands.

```
   1 REM ************************
         *Demo Program, type in*
         *machine code first ! *
         ************************
   5 CLEAR 64899: LOAD "6 Comman
ds"CODE
  1Ø LET a=USR 649ØØ: GO TO 9ØØØ
  15 REM Change values in lines
5 and 1Ø for use in 16K
  2Ø RESTORE
  3Ø FOR f=Ø TO 7: FOR g=Ø TO 7
  4Ø PRINT AT f*2,g*4; INK f; PA
PER g;"; FLASH 1;"; BRIGHT 1
;"; FLASH Ø;"
  5Ø PRINT AT f*2+1,g*4; PAPER f
; INK.g;"; FLASH 1;"; BRIGHT
 1;"; FLASH Ø;"
  6Ø NEXT g
  7Ø READ a$: PRINT  INK 8; PAPE
R 8; FLASH 8; BRIGHT 8;AT f*2,Ø;
"INK=";a$;AT f*2+1,Ø;"PAPER=";a$
  8Ø NEXT f
  9Ø PRINT AT 17,Ø; INK 1;"1st &
 4th Numbers:"'"Ø=none,1=BR,2=FL
,3=FL+BR,8=all"' INK 2;"Others:
Ø to 7=Black to White,  8=All. A
lso 9=Contrast for 5&6th"
 1ØØ INPUT "ATTR ";f;",";p;",";i
;"  TO ";f1;",";p1;",";i1
 11Ø IF f<Ø OR p<Ø OR i<Ø OR f1<
Ø OR p1<Ø OR i1<Ø THEN  GO TO 1Ø
Ø
 12Ø IF (f>3 AND f<>8) OR p>8 OR
 i>8 OR (f1>3 AND f1<>8) OR p1>9
 OR i1>9 THEN  GO TO 1ØØ
 15Ø FOR g=Ø TO 31: PRINT AT 21,
g; INK i-(INT (RND*8) AND i=8);
BRIGHT INT (RND*2); FLASH INT (R
ND*2); PAPER p-(INT (RND*8) AND
p=8);"&": NEXT g
 18Ø PRINT #1;AT Ø,Ø; INK 3;"Pre
ss any key to demonstrate:";AT 1
,Ø; INK Ø;"ATTR "; INK 1;f;",";
INK 2;p;",";i;"  TO "; INK 1;f1;
",";  INK 2;p1;",";i1: PAUSE Ø
 2ØØATTR f,p,i TO f1,p1,i1
 25Ø POKE AT 23296,"Ø6Ø1 CDØØØE
C9": REM LD B,+1 CALL ØEØØ RET
 26Ø LET a=USR 23296: PRINT #1;A
T 1,Ø; INK 9;"Press any Key(exce
pt ~i~)to Cont"
 3ØØ PAUSE Ø: BEEP .1,1Ø: BEEP .
1,15: BEEP .1,2Ø: BEEP .1,12: IF
 PEEK 2356Ø<>CODE "i" THEN  GO T
O 2Ø
1ØØØ DATA "Black","Blue","Red","
Magenta","Green","Cyan","Yellow"
,"White"
9ØØØ REM Instructions
9Ø1Ø CLS : PRINT  INK 3;AT Ø,6;"
Six Extra Commands";AT 1,7;"by P
aul Matthews"
9Ø2Ø PRINT AT 3,Ø;"This 44Ø Byte
 routine gives the user the foll
owing new Commands:"
9Ø3Ø INK 1: PRINT AT 6,Ø;"PLOT *
x,y "; INK 2;"which can PLOT eve
n onthe bottom two lines."
9Ø4Ø PRINT AT 9,Ø;"POKE *addr,nu
m "; INK 2;"which will POKE a16-
bit number into an address."
9Ø5Ø PRINT AT 12,Ø;"POKE AT addr
,h$ "; INK 2;"which will POKE ad
dr with hex held in the string"
9Ø6Ø PRINT AT 15,Ø;"ATTR f,p,i T
O nf,np,ni "; INK 2;"which    se
eks out attributes and changespa
rt or all of each found."
9Ø7Ø PRINT AT 19,Ø;"SIN "; INK 2
;"i.e. ~Screen INvert~"
9Ø8Ø PRINT AT 2Ø,Ø; INK Ø;"and";
 INK 1;AT 21,Ø;"INVERSE "; INK 2
;"i.e. ~Attribute Inverse~"
9Ø9Ø PRINT #1; INK 3;"Press any
key for Demonstration"
91ØØ PAUSE Ø: INPUT ""
911Ø POKE AT 23296,"Ø615 CDØØØE
C9": REM LD B,+15 CALL ØEØØ RET
912Ø FOR f=Ø TO.11: LET a=USR 23
296: NEXT f
913Ø INK Ø: PRINT AT 11,Ø;"SIN e
ffects the POINT command aseach
pixel changes its state"'"e.g. P
OINT (Ø,Ø) = ";POINT (Ø,Ø)
914Ø PRINT #1; INK 3;" Press any
 key for Demo of SIN": PAUSE Ø:
INPUT ""
915ØSIN : PRINT AT 14,Ø;"and now
 POINT (Ø,Ø) = ";POINT (Ø,Ø)
916Ø PAUSE 5Ø: PRINT AT 16,Ø;"IN
VERSE effects ATTR as all INK an
d PAPER colours are swopped    e.
g. ATTR (Ø,Ø) = ";ATTR (Ø,Ø)
917Ø PRINT #1; OVER 1; INK 3;" P
ress any key for INVERSE Demo ":
 PAUSE Ø: INPUT ""
918Ø INVERSE : PRINT AT 19,Ø;"an
d now ATTR (Ø,Ø) = ";ATTR (Ø,Ø)'
"(but still POINT (Ø,Ø) = ";POIN
T (Ø,Ø);")"
919Ø PRINT #1; INK 3;AT Ø,Ø;"  P
ress any key for ATTR Demo    (N
ote:To stop demo press ~i~)   "
92ØØ PAUSE Ø: CLS : GO TO 2Ø
9999 SAVE "~6~ Demo" LINE 1
```

·DEACTIVATORS·

Deactivators
Ariolasoft
£8.95



As chief of security at a top secret research centre, you are already in considerable hot water as a group of terrorists have bypassed your security system and planted a series of bombs, timed to detonate in sequence over a short period of time. As if that wasn't enough, they have also reprogrammed the guards to attack everything on sight. Your only hope of regaining some of your lost credibility is to send in a team of deactivating droids and get rid of the bombs as quickly as possible.

The building is on five levels and each one must be cleared in turn. At the bottom of the screen is a map showing the layout of the floor together with the location of your droids and the bombs. A series of icons can be used to select a specific droid and also to view any two adjacent rooms. These are drawn in 3-D perspective and show details of all the exits and objects present.

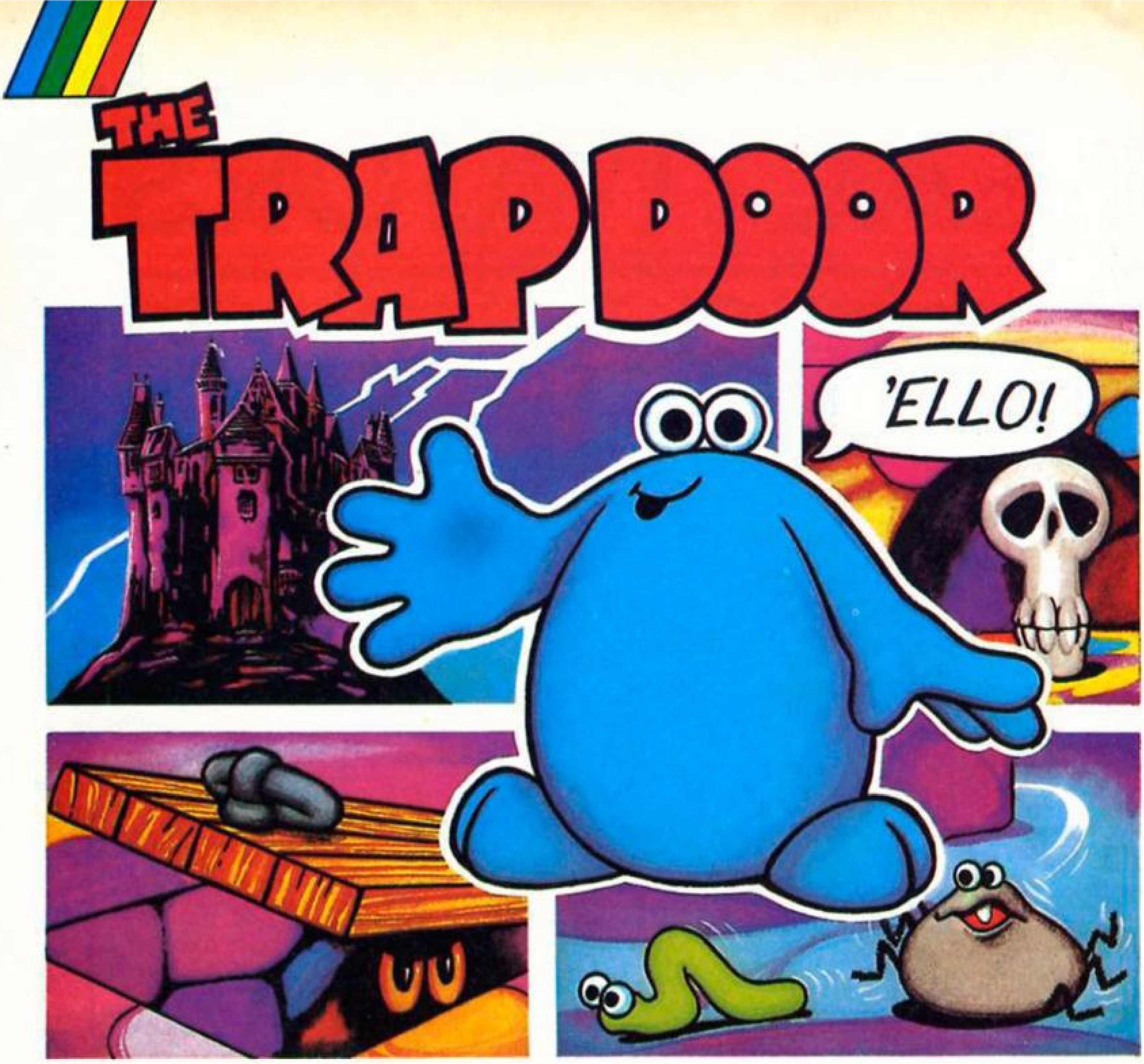
Clearing the level of bombs is no easy matter and there are several problems confronting the droids. Not the least of these is that they are limited in which rooms they have access to and this results in one droid having to pick up a bomb and throw it through a window into a different sector. It helps if you have another droid there waiting to catch it for there is always the chance that the bomb will detonate prematurely. You will also find circuit boards lying around and the object with these is to return them to the computer room. These are essential as they open up extra windows and doors, activate teleports and remove force fields. Your aim is to find a room at the corner of the building with a window leading to the outside through which you can dispose of the bomb.

On top of all this there are the guards to contend with. They will destroy you on contact and the only way to neutralise them is by persuading them to follow you through a hole in the floor onto the next lower level. Do this often enough and the guards will blow up. Still your problems aren't over. Because of the nature of the research establishment, some of the rooms have decidedly funny properties. The main difference you will notice is that they are all colour coded and this refers to the gravity level in the room. This ranges from ½G to 3G and affects your aim badly when you try to throw objects out of a room. At higher levels, rooms are rotated through 90 or 180 degrees, giving rise to another set of problems. Or a room may be blacked out totally.

Deactivators is an excellent mixture of strategy and action. It is not easy to work out the correct sequence of events that you must organise and a lot of people will find that they are getting nowhere fast with the game. Certainly it is very daunting at first and it would help considerably if you could get a decent demonstration before deciding if it is the game for you. Love it or hate it, it will definitely give your grey matter a good working over.







What is the Bad Tempered Thing, how do you boil Slimies, and where does Terry Wogan fit in to all this? The answer's beneath the Trap Door.

The Trap Door
Piranha
£7.95

Not that long ago, Programmer Don Priestley wrote a computer game based on the Popeye comic strip and cartoon series, and for the game he came up with some new techniques that created some of the largest coloured sprites ever seen on the Spectrum, as well as masking techniques that almost made you forget about the Speccy's attribute problems. Unfortunately, the Popeye game didn't set the world aiight (though I thought it was excellent), but now Don Priestley and his mega-sprites have reappeared at the helm of the first game from the new Piranha label, The Trap Door, based on a new children's series produced by Terry Wogan's own production company, and due to be seen on television this autumn.

Berk and Drut

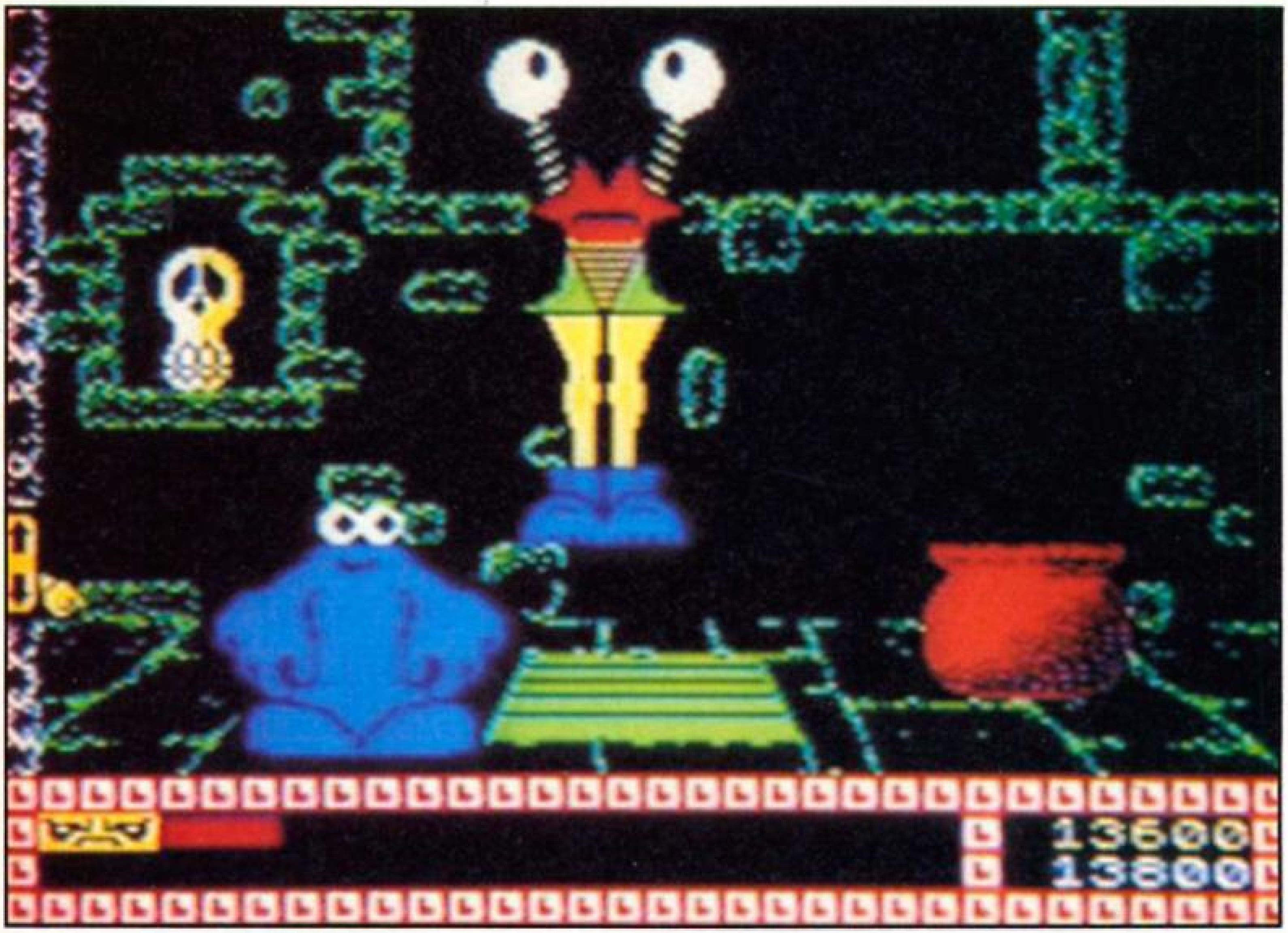
The main character in the series is a blue blob-like fellow by the name of Berk. Berk is the servant of the Bad Tempered Thing who lives in the upper regions of the castle whilst Berk rushes around in the castle cellar doing the Thing's bidding (which normally means getting his meals — worms, boiled slimies, that sort of thing).

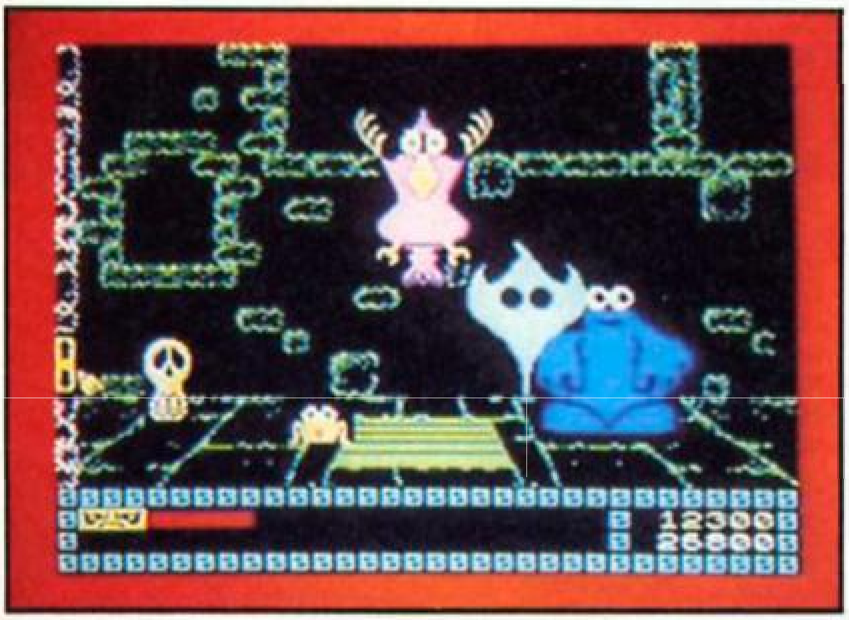
Also tucked away in the cellar is the Trap Door, beneath which lurk all sorts of weird creatures. Some of these things can actually help Berk in his chores, though others are just plain awkward.

In the game, you are in control of Berk and have to complete the tasks that will get shouted down to you from upstairs. This involves looking for, and also making use of the creatures that will pop up from the trap door just about every time you open it. Along the way you'll have to solve some of the most devious and imaginative problems that you're likely to come across in an arcade/adventure. What, for instance, do you do with the bullet if you haven't got a gun and how do you make the bird lay eggs? Fortunately Berk has a friend called Boni, a disembodied skull who can provide the odd bit of advice if you're not sure what to do next.

Down in the dungeon

The playing area down in the cellar isn't all that huge, consisting of barely a dozen screens, but, as with the Popeye game, one of the things that makes the game so enjoyable is the 'layered' effect that allows you to move characters and objects not just up/down and left/right in two dimensions, but also into and out of the screen to create a real 3D effect. So,







although the number of individual screens is quite limited, there is a lot more that can be done than in most other games. There are passages and staircases that you can wander through, and a number of objects that need to be positioned carefully in order to complete your tasks.

Sprites

But the main feature of Trap Door is probably the quality of the graphics and animation. Berk himself is a sprite of not inconsiderable proportions, but some of the monsters that pop up out of the trap door are larger than anything that I've ever seen drawn on a Spectrum screen, and yet they move around the screen smoothly and without any attribute problems. The animation is so good that it's fun to just sit and watch all the action in front of you. One of the nice things about the television program is the way that all sorts of little creepy crawlies just wander about doing odd things irrespective of what the main characters are up to, and this has been carried over into the computer game so that you can have Drut (a small yellow blob) chasing worms around the screen while Berk is busy trying to do something else. This adds to the atmosphere of the game and makes it feel almost like a real cartoon world.

Don Priestley's earlier experiments with this type of game haven't been wildly successful, perhaps they fall rather awkwardly between the two types of arcade and adventure game (and also because the animation techniques are so costly in memory that a 48K doesn't really do them justice). But hopefully with the publicity surrounding the TV series people will take a close look at Trap Door and enjoy it as much as I did.

MONSTER HIT

CYRUS II



Chess fanatics have never had it so good as another strong contender for the Spectrum grandmaster crown enters the arena.

Cyrus II Mark I
Alligata
£11.95

Cyrus II arrives on the Spectrum with a good reputation from its performance on other micros and with the recent release of Psi Chess and Colossus 4 Chess, Spectrum owners are being spoilt for choice.

There will obviously be a lot of debate on which is the strongest program available and Alligata have thrown down the gauntlet with a statement from their managing director included on the inlay which says, "I am confident this game will beat all existing Sinclair Spectrum chess programs published."

No doubt the makers of Colossus 4 Chess will dispute this. Making a comparative assessment of the two programs with any degree of accuracy by playing them off against each other requires an awful lot of time — too much to fall within the parameters of this review. I'm sure both software houses will be letting us know the results soon enough.

Leaving aside the issue of which program can beat the rest, it has to be said the Cyrus II is a very strong player and there are all the features included which single it out as an excellent opponent.

There are no 3-D graphics (the Mark 2 version for the 128 will have them) but the screen display is very bright and clear. There are nine standard levels of play with thinking time ranging from two seconds to three and a half minutes. In addition there is an adaptable mode (Cyrus adapts to your response time), infinite mode (Cyrus will wait until you decide he can move) and a problem mode (Cyrus will search for a checkmate in five moves or less).

Other features include replaying the game, taking back moves, playing both sides, changing sides, hints from Cyrus and demonstration games. There is also a simple print-out facility, if you want to record game situations for posterity.

One big advantage with Cyrus II is that accessing different modes is very simple with most commands being made with a single key entry.

Making the moves is also very straight forward by use of cursor keys (there is no joystick option). Simply move the cursor key to your chosen piece, hit enter, then move a second flashing cursor to your chosen square. Hit enter again and the move is made.

This program is a very welcome addition to the Spectrum chess scene and is recommended highly, the drawback however is the price. At £11.95 it's two quid more expensive than both Psi Chess and Colossus 4 and one wonders why it couldn't have been released at just under a tenner, like its rivals.

Cyrus II is a superb program and still worthy of a Monster Hit at the price but in a sector of the market which is now fiercely competitive it may have spoilt its chances with those who are after just one chess game.

MONSTER HIT



MONITOR

If you're writing your own machine code programs you'll need a good monitor to help debug them — so here it is.

BY JON ELLIS

Over the course of writing many machine code routines I realised that I had no good program for debugging them. So, if you haven't got a program – write it yourself. Here is that program, a monitor which should provide all the functions you would normally need in the course of debugging a machine code program.

Numbers can be entered in either hex or decimal, regardless of the state of the number base toggle. Decimal numbers must all be five digits long, being bulked out with leading zeros if necessary. To enter a number in hex, type '&' as the first character, and then the number in four hex digits, again using leading zeros if necessary. If only one byte is required then the low byte is taken, ie if you enter a number greater than 255 when the Monitor is expecting a number in the range 0 to 255, then your input will be taken as modulo 256.

After most functions have been executed the Monitor will pause for you to examine the results of the function, if any. To terminate this pause press either 'X' or BREAK and you will then be returned to the menu.

The Monitor is written so that you can treat the Spectrum as a Z80 processor only as far as possible. This means that, although the Spectrum operating system requires the iy register to have the value 23610 at all times, unless your routine is going to use the operating system, you may ignore this requirement. A similar situation occurs with the h'l' pair. The registers that are used when your routine is run are not those used by the Monitor or the Spectrum ROM.

To abort a function you should press BREAK which will return you to the menu.

Monitor Functions

The monitor offers a menu of 18 functions:

1) *Alter memory.* This function will prompt for a start address (see note on the input of numbers), and the program will then print the address and its contents. The Monitor will then wait for you to input the new value before moving on to the next byte. When you have filled up a screen, press 'X' for another page. At any time you may press BREAK to escape.
2) *Breakpoint.* This function enables you to place an instruction in the middle of the code that is being debugged. Which causes execution to be stopped, and control returned to the Monitor so that you can alter/inspect the registers etc. The breakpoint instruction is three bytes long, and thus two breakpoints should not be inserted less than four bytes apart otherwise a crash might occur on execution. The code that occupied the three bytes overwritten by the breakpoint instruction is stored, and will be replaced when the breakpoint is deleted. When you select this function from the menu it will prompt for the address at which the breakpoint is to be inserted. A total of nine breakpoints may be used at once. If all nine are in use then nothing will happen when the function is selected.
3) *Convert number.* This function will prompt for the input of a number and then print the number in decimal, hex, and binary.
4) *Delete breakpoint.* When selected, this command will display all nine of the breakpoints, and then ask for the number of the breakpoint to be deleted. (Note that you should enter the number of the breakpoint (1-9) *not* its address). The three bytes that were overwritten will then be replaced.
5) *Examine stack.* The Monitor provides 40 bytes of stack space solely for use by the object program. On enterting the Monitor or resetting the registers, the stack is reduced to one word in length, this word being a return address inside the Monitor, which should prevent a crash in the vent of a RET instruction being executed at the end of the object program. On selection, this function displays all the words currently on the user stack.
6) *Fill memory block.* This command allows you to fill a block of memory with a specified value. It prompts for a start address, the length of the block to be filled, and the byte to be used.
7) *Jump to routine.* The function will prompt for the address to be jumped to, and execution will continue from this address using the values of the user registers (see later). This function is to allow for the object code to be tested. Ideally a breakpoint should be used at the end of the code to return to the Monitor, in which case the report 'BREAK at (address)' will be displayed. However a RET instruction should also work providing that the routine has used the stack correctly.
8) *Look at breakpoints.* This function displays the addresses of each of the nine breakpoints, an address of 0 indicating that the breakpoint is unused.
9) *Move memory block.* This command allows you to copy blocks of memory to other areas. It prompts for the start address of the block, the start of the destination area, and the length of the block to be copied.
10) *Number base toggle.* This toggle determines the base in which all numeric output from the monitor will be displayed. It toggles between hex, decimal and binary. Only some output can be displayed in binary due to the length of binard word (16 characters), other output being displayed in hex. The toggle defaults to decimal.
11) *One-step routine.* This is probably one of the most useful functions. It enables you to step through the object code one instruction at a time, whilst keeping track of the user register values. The Monitor will prompt for a start address, and will commence stepping from that address, updating the register display after each instruction and then pausing for you to inspect the registers. To move to the next instruction, press 'X'. To escape, press BREAK.
12) *Printers toggle.* When toggled to ON, this causes output from the View memory function to be dumped to the ZX Printer. The toggle defaults to OFF.

▶ ▶ ▶

Instructions

The Monitor is written entirely in machine code and occupies the memory from 28350 to 32767, and is thus 4418 bytes long. This means that it will fit on a 16K Spectrum, but it is unlikely that the expansion opportunities offered by the User program feature will be utilised fully.

Type in listing 1 and save it — this is the hex loader. Run the program and you will be prompted for the string of hex digits and then for the check digit. Although the check digit might not look right it is actually more effective than a normal checksum, spotting the vast majority of transposition errors. When you have finished the program will save the finished code and then verify it.

Reset the machine and then type **CLEAR 28349: LOAD "MONITOR"CODE** as a direct command. After the code has loaded you should enter the Monitor with RANDOMIZE USR 28350. A menu of the 18 functions should be displayed. You should test each of the functions carefully. If you discover any errors then listing 3 should help — it displays the code you have entered in the same format as listing 2.

13) *Quit.* Returns control to BASIC.
14) *Register display.* This function displays the values of the user registers, both the normal and the alternate set. The current values of the interrupt vector register, i, and the memory refresh register, r, are also displayed. The value given for the program counter, pc, refers to the last instruction executed. The value of the stack pointer, sp, that is displayed refers to the user stack. The flags register of the normal af pair is shown expanded into bit format, with each of the six flags labelled. The value of the maskable interrupt flip-flop is also shown, the interrupts being enabled or disabled as appropriate when the user code is executed.
15) *Specify entry values.* This function allows you to change the values of the user registers, perhaps to test a subroutine whose parameters are passed to it in the registers. You can also specify the value of the zero flag and the carry flag. You should select the register to be altered by pressing the appropriate letter. To escape press BREAK.
16) *Use other program.* This function allows you to use another program and to call it from the Monitor. The function is not designed for the execution of the object program, but for the calling of some other utility such as a disassembler. When you quit the other program, control will return to the Monitor. The use of this function to call an assembler forms a very powerful development tool.
17) *View memory.* This displays the contents of memory from the specified start location, dumping also to the printer if the printer toggle is on. The routine displays 110 bytes per page, 20 bytes per page if using binary. When the page is full the computer will wait for you to press 'X' before proceeding to the next page. To escape, press BREAK.
18) *Zero registers.* This function restores the user registers to their original values that they held when the Monitor was first entered. In most cases this is 0, but the iy register defaults to 23610 and the h'l' pair defaults to 10072 as required by the Spectrum operating system. The user stack is also reset and preloaded with the return address mentioned earlier in the examine stack function.

Listing 1

Hex loader program.

```
  10 REM ************************
  20 REM **    HEX  LOADER     **
  30 REM ** by Jon Ellis 1985 **
  40 REM ************************
  50 CLEAR 28349
  60 FOR F=28350 TO 32767 STEP 8
  70 PRINT F;
  80 LET CHECK=0
  90 INPUT "Enter bytes:"; LINE
H$
 100 IF LEN H$<>16 THEN  PRINT "
Error in length": BEEP .2,0: GO
TO 70
 110 PRINT TAB 8;H$;
 120 FOR G=1 TO 8
 130 LET BYTE=16*(CODE H$(1)-48-
(7 AND H$(1)>"9"))+(CODE H$(2)-4
8-(7 AND H$(2)>"9"))
 140 POKE F-1+G,BYTE
 150 LET CHECK=CHECK+BYTE*G
 160 LET H$=H$(3 TO )
 170 NEXT G
 180 INPUT "Enter check digit:";
 LINE C$
 190 IF LEN C$<>1 THEN  GO TO 18
0
 200 PRINT TAB 27;" ) ";C$
 210 LET DIGIT=CHECK-9*INT (CHEC
K/9)
 220 IF DIGIT<>VAL C$ THEN  PRIN
T "Error in string - input again
": BEEP .2,-12: GO TO 70
 230 NEXT F
 240 PRINT  FLASH 1;"READY TAPE
FOR SAVING"
 250 SAVE "MONITOR"CODE 28350,44
18
 260 CLS
 270 PRINT  FLASH 1;"READY TAPE
TO VERIFY"
 280 VERIFY "MONITOR"CODE 28350,
4418
```

Listing 2

Hex data for the machine code.

```
28350   CDD777CD0678DD21   ) 8
28358   A27FDDCB00960175   ) 8
28366   0111207BCD3C2006   ) 0
28374   1221435801120036   ) 5
28382   3A1910FB3E16D73E   ) 3
28390   0DD73E16D7FD3655   ) 3
28398   F2DDCB006620083E   ) 3
28406   4FD73E4ED7180DFD   ) 3
28414   3655673E4FD73E46   ) 4
28422   D73E46D73E16D73E   ) 0
28430   0BD73E19D73AA47F   ) 3
28438   FE102822FE02280F   ) 1
28446   FD36557B3E44D73E   ) 2
28454   45D73E43D7181CFD   ) 0
28462   3655683E42D73E49   ) 2
28470   D73E4ED7180DFD36   ) 0
28478   55703E48D73E45D7   ) 5
28486   3E58D7FD365538CD   ) 0
28494   AB7701120021227A   ) 6
28502   EDB120EF11227AA7   ) 0
28510   ED52282911347A19   ) 3
28518   5E2356EBCD706FC3   ) 5
28526   C16EE9CD06783E41   ) 0
28534   D7CD8479D822AA7F   ) 0
28542   3E0DD7FD7E4FFE02   ) 8
28550   2007CDA078D8CD06   ) 0
28558   782AAA7FE52AA37F   ) 6
28566   22A87F3E02BCCCD1   ) 8
28574   71DDCB00CEE1E5CD   ) 1
28582   1C782AA87F22A37F   ) 3
28590   3E17D73E0AD73E20   ) 7
28598   D7E1E57E2322AA7F   ) 4
28606   6F2600DDCB008ECD   ) 4
28614   1C783E17D73E14D7   ) 1
28622   3E20D7CDAD787DE1   ) 1
28630   D87718A43AA77FFE   ) 4
28638   09D0CD06783E42D7   ) 6
28646   CD7379D822AA7F3A   ) 2
28654   A77F3C32A77F3D57   ) 2
28662   8782825F160021F8   ) 7
28670   7A19ED5BAA7F7323   ) 2
28678   72231A773ECD1223   ) 2
28686   131A77015F777912   ) 2
28694   23131A777812C9CD   ) 8
28702   06783E43D7CD8479   ) 1
28710   D822AA7F2AA37FE5   ) 0
28718   010E0011457FCD3C   ) 5
28726   20D52AAA7F11050A   ) 1
28734   ED53A37FCD1C7801   ) 3
28742   0900D1CD3C20D511   ) 2
28750   0410ED53A37F7CFE   ) 3
28758   00DDCB00CE2004DD   ) 1
28766   CB008ECD1C78010C   ) 0
28774   00D1CD3C20111002   ) 7
28782   ED53A37FCD1C78E1   ) 4
28790   22A37FC3A078DDCB   ) 2
28798   00D6CD27713E16D7   ) 5
28806   3E15D7AFD73E44D7   ) 7
28814   CD8479D87DFE0A30   ) 6
28822   E58787856F260011   ) 6
28830   F37A195E23562301   ) 1
28838   0300EDB0AF2B772B   ) 2
28846   772B772B772B77DD   ) 3
28854   360500012D001100   ) 7
28862   5B21F87AE5EDB00E   ) 8
28870   2DE11E00CD1F7106   ) 7
28878   0911F87A21005BC5   ) 8
28886   E50605AFBE232004   ) 5
28894   10FA180AE1E50105   ) 6
28902   00EDB0DD3405E123   ) 8
28910   23232323C110E001   ) 8
28918   2D0021005B1E0018   ) 8
28926   20CD06783E46D7CD   ) 0
28934   8479D822AA7FCDA6   ) 7
28942   79D822AC7FCDB779   ) 3
28950   D85D2AAA7FED4BAC   ) 0
28958   7F73230B78B120F9   ) 1
28966   C9CD0678011B0011   ) 8
28974   687FCD3C20060921   ) 8
28982   F87A3E0A90C630D7   ) 2
28990   C501080011837FCD   ) 5
28998   3C205E2356232323   ) 1
29006   23E5EBDDCB00CECD   ) 5
29014   1C783E0DD73E0DD7   ) 2
29022   3E17D73E07D7AFD7   ) 6
29030   E1C110CEDDCB0056   ) 3
29038   C0C3A078CD06783E   ) 6
29046   4DD7CD8479D822AA   ) 1
29054   7FCD9579D822AC7F   ) 4
```

MONITOR

```
29062  CDA679D8444DED5B  ) 3
29070  AC7F2AAA7FED80C9  ) 0
29078  DD7E00EE10DD7700  ) 1
29086  C9CD06783E55D7CD  ) 7
29094  7379D8CD706FDD21  ) 2
29102  A27FFD213A5CD921  ) 8
29110  5827D9C3A078DDCB  ) 6
29118  02662018DDCB025E  ) 4
29126  2009DD36020ADD36  ) 8
29134  0105C9DD360210DD  ) 6
29142  360104C9DD360202  ) 0
29150  DD360110C9CD0678  ) 6
29158  2AA37F22A87F7CFE  ) 5
29166  02200621041022A3  ) 2
29174  7F01FC0011E37CCD  ) 7
29182  3C2021AE7F11DF7D  ) 5
29190  060CC578FE032005  ) 7
29198  2323C110F5010300  ) 2
29206  CD3C20D55E235623  ) 0
29214  E5EBDDCB00CECD1C  ) 2
29222  78E1D1C110DC3E16  ) 1
29230  D73E04D73E08D72A  ) 8
29238  A87F22A37F3AC17F  ) 8
29246  6F2600DDCB000ECD  ) 4
29254  1C783AA97FFE0220  ) 1
29262  0621041022A37F3E  ) 8
29270  16D73E10D73E08D7  ) 1
29278  ED576F2600DDCB01  ) 6
29286  8ECD1C783E16D73E  ) 4
29294  10D73E18D7ED5F6F  ) 8
29302  2600DDCB018ECD1C  ) 6
29310  782AC07F3E16D73E  ) 5
29318  13D73E07D7060878  ) 0
29326  FE06280DFE042809  ) 5
29334  AFCB2517C630D718  ) 6
29342  053E78D7CB253E20  ) 0
29350  D710E42AA87F22A3  ) 4
29358  7F3E16D73E15D73E  ) 5
29366  08D73E4FD73AC87F  ) 3
29374  FE0028083E4ED73E  ) 8
29382  20D718063E46D73E  ) 4
29390  46D7DDCB0056C0C3  ) 4
29398  A078CD067821FF7F  ) 3
29406  ED5BC27FA7ED52C8  ) 0
29414  01110011467ECD3C  ) 1
29422  2045CB38483E14B8  ) 8
29430  D821FE7F3E0DD779  ) 3
29438  90C5E5DDCB000626  ) 1
29446  006FCD1C783E17D7  ) 4
29454  3E0AD7AFD7E1DDCB  ) 3
29462  00CEE5CD1C78E156  ) 2
29470  2B5E2BE5EB3E17D7  ) 6
29478  3E14D7AFD7CD1C78  ) 3
29486  E1C110C8C3A078CD  ) 6
29494  06783E4AD7CD7379  ) 0
29502  D822CA7FCD0678ED  ) 2
29510  73C67FED7BC27FDD  ) 3
29518  CB00562004218E73  ) 3
29526  E53AC87FFE002001  ) 3
29534  F3DD2ABC7FFD2ABE  ) 4
29542  7F2AC07FE5F1ED4B  ) 7
29550  AE7FED5B807F2AB2  ) 3
29558  7F08D92AB47FE5F1  ) 8
29566  08ED4BB67FED5BB8  ) 3
29574  7F2ABA7FD9C3C97F  ) 6
29582  22B27FED73C27FED  ) 6
29590  7BC67FF5E122C07F  ) 7
29598  ED43AE7FED53B07F  ) 4
29606  DD22BC7FFD22BE7F  ) 1
29614  D9ED43B67FED53B8  ) 6
29622  7F22BA7F215827D9  ) 3
29630  08F5E122B47F08FD  ) 2
29638  213A5CDD21A27FFB  ) 6
29646  DDCB0056C0C3A078  ) 6
29654  CD06783E4FD7CD73  ) 5
29662  79D822C47FED5BC4  ) 4
29670  7FDD3603001AFEDD  ) 8
29678  2830FEFD282CFEED  ) 0
29686  281AFECB28682150  ) 5
29694  7A011800EDB1285E  ) 1
29702  21707A011A00EDB1  ) 7
29710  28511855131A218A  ) 4
29718  7A010000EDB12840  ) 1
29726  1844131AFEED281A  ) 0
29734  FECB282221927A01  ) 2
29742  0400EDB1282A2196  ) 3
29750  7A011A00EDB12823  ) 3
29758  1824131AFE632815  ) 4
29766  FE6B281118151313  ) 8
29774  1A21B07A011F00ED  ) 0
29782  B128051806DD3403  ) 6
29790  DD3403DD3403DD34  ) 3
29798  03DD3403060521D1  ) 2
29806  7F36002310FBED4B  ) 8
29814  A57F06002AC47F11  ) 4
29822  D17FEDB022C47F11  ) 1
29830  D17F1AFEE920092A  ) 1
29838  B27F22C47FC3CB75  ) 8
29846  FEDD2010131A1BFE  ) 7
29854  E9201D2ABC7F22C4  ) 1
29862  7FC3CB75FEFD2010  ) 0
29870  131A1BFEE920092A  ) 8
29878  BE7F22C47FC3CB75  ) 2
29886  1AFEED200C131A1B  ) 1
29894  FE45CA4B75FE4D28  ) 0
29902  7C1AFEF3CAB475FE  ) 2
29910  FBCABA7521CF7A01  ) 8
29918  2900EDB1C2BE75FE  ) 4
29926  C32853FECD2824FE  ) 4
29934  18CA7D75FEC92855  ) 0
29942  FE10CA9D75CB7F28  ) 5
29950  77E6072842FE0728  ) 4
29958  61FE02282BCDDD79  ) 6
29966  D2CB75CDC879ED5B  ) 5
29974  C47FED53AA7F2AD2  ) 3
29982  7F22C47F115F77A7  ) 2
29990  ED52C2CB752AC27F  ) 2
29998  232322C27FC39077  ) 5
30006  CDDD79D2CB752AD2  ) 6
30014  7F22C47FC3CB75CD  ) 3
30022  DD79C2CB75ED73C6  ) 7
30030  7FED7BC27FE1ED73  ) 1
30038  C27FED7BC67F22C4  ) 7
30046  7F118E73A7ED52C8  ) 6
30054  1863CDC8791AE638  ) 1
30062  6F260022C47F1855  ) 6
30070  E618CDDD79304E3A  ) 8
30078  D27FCB7F28105FAF  ) 0
30086  57935F2AC47FA7ED  ) 7
30094  5222C47F18375F16  ) 8
30102  002AC47F1918F2F5  ) 0
30110  ED4BAE7F2AC07FE5  ) 1
30118  F105F5E122C07FED  ) 8
30126  43AE7FF118C2AF32  ) 7
30134  C87F18113E0118F7  ) 3
30142  DDCB00D621D17F22  ) 4
30150  CA7FCD4573CDE371  ) 2
30158  DDCB0096CDA078D8  ) 2
30166  C3E373CDBE77D8FE  ) 2
30174  59C0ED7B3D5C3B38  ) 7
30182  C9CDBE77D8FE59C0  ) 3
30190  CDD777C9CD067811  ) 8
30198  577E01D800CD3C20  ) 7
30206  11AE7F060E1A6F26  ) 7
30214  00DDCB008ED5C5CD  ) 1
30222  187ACD1C78C1D113  ) 3
30230  10EBDDCB00CE2ABC  ) 0
30238  7FCD187ACD1C78CD  ) 0
30246  187A2ABE7FCD1C78  ) 8
30254  DDCB008ECD187A3A  ) 1
30262  C17F6F2600CD1C78  ) 6
30270  CD187A3AC07FE601  ) 6
30278  6F2600CD1C78CD18  ) 2
30286  7A3AC07F0707E601  ) 2
30294  6F2600CD1C783E15  ) 3
30302  D7AFD7CDAB77D8FE  ) 6
30310  4138F8FE5430F4D6  ) 8
30318  401100586F260029  ) 4
30326  292929291936C4F5  ) 6
30334  01130011327FCD3C  ) 1
30342  20CDAD78D1D87AFE  ) 8
30350  0F3837FE11283FFE  ) 2
30358  13281CFE122810FE  ) 1
30366  10280622BC7FC3F2  ) 6
30374  7522BE7FC3F2757D  ) 1
30382  E6016F16FE18087D  ) 0
30390  E60116BF0F0F6F3A  ) 2
30398  C07FA2B532C07FC3  ) 8
30406  F2755F16007D21AD  ) 1
30414  7F1977C3F2757D32  ) 1
30422  C17FC3F275CD0678  ) 2
30430  3E56D7CD8479D822  ) 0
30438  AA7FCD06780616C5  ) 0
30446  78DD77043E16D73E  ) 3
30454  16DD9604D7AFD72A  ) 0
30462  AA7FDDCB00CECD1C  ) 6
30470  7806053E16D73E16  ) 2
30478  DD9604D73E05904F  ) 7
30486  878781C607D73AA4  ) 1
30494  7FFE02200C3E16D7  ) 6
30502  3E16DD9604D73E17  ) 3
30510  D72AAA7F7E2322AA  ) 2
30518  7F6F2600DDCB008E  ) 7
30526  CD1C783AA47FFE02  ) 8
30534  280210BFCD541FC1  ) 6
30542  D8109CDDCB0066C4  ) 2
30550  AC0ECDA078D8C3E8  ) 2
30558  7622B27FE122AA7F  ) 2
30566  ED73C27FED7BC67F  ) 5
30574  219077E5ED73C67F  ) 3
30582  ED7BC27F2AB27FDD  ) 0
30590  22BC7FDD21A27FDD  ) 6
30598  CB00D6DD2ABC7FC3  ) 8
30606  8E73CD06782AAA7F  ) 1
30614  2B2B2B22C47F0117  ) 5
30622  00118B7FCD3C20CD  ) 5
30630  1C78C3A078FDCB01  ) 0
30638  AE76FDCB016E28F5  ) 8
30646  CD541F3A085C3FC9  ) 0
30654  012300119D7CCD3C  ) 1
30662  20CDAB77D8F50123  ) 2
30670  0011C07CCD3C20F1  ) 8
30678  C9061621AE7F3600  ) 0
30686  2310FBED73C67F31  ) 6
30694  FF7F218E73E5ED73  ) 7
30702  C27FED7BC67F3E01  ) 2
30710  32C87F21582722BA  ) 6
30718  7F213A5C22BE7FC9  ) 6
30726  FD3653303E07CD9B  ) 0
30734  22CD6B0D3E02CD01  ) 8
30742  16FDCB30DEC9E5C5  ) 2
30750  DDCB0086DD460178  ) 1
30758  3DC5E5CD8578EBE1  ) 6
30766  C10E000CA7ED5230  ) 1
30774  FA190D202778FE01  ) 0
30782  2822DDCB0046201C  ) 3
30790  DD7E02FE0A2823DD  ) 0
30798  CB004E200FFE1078  ) 5
30806  2006FE0338061812  ) 6
30814  FE09300EDDCB00C6  ) 5
30822  79C630FE3A3802C6  ) 4
30830  07D710B33AA47FFE  ) 5
30838  0A2809FE103E6828  ) 5
30846  023E62D7C1E1C9FE  ) 5
30854  00210100C8473AA4  ) 0
30862  7F110000626FC547  ) 6
30870  051910FD545DC110  ) 8
30878  F5C9FDCB30DECDAB  ) 5
30886  77D8FE58C818F33A  ) 6
30894  A47F32A87FDDCB00  ) 5
30902  9E21CC7F0605CDAB  ) 4
30910  77D8FE0C201C78FE  ) 7
30918  0528F3FE042004DD  ) 4
30926  CB009E3E08D73E20  ) 2
30934  D73E08D72B0418DE  ) 4
30942  18DCFE262816FE30  ) 2
30950  38D4FE3A3818DDCB  ) 2
30958  005E28CAFE4138C6  ) 0
30966  FE47380ADDCB005E  ) 7
30974  28BCDDCB00DE7723  ) 8
30982  D7108321CC7F7EFE  ) 5
30990  2611000028263E0A  ) 6
30998  32A47FE5DDE10605  ) 0
31006  C5783DD5CD8578D1  ) 2
31014  EBDD7E00D6302804  ) 7
31022  193D18FAEBDD23C1  ) 1
31030  10E6182D3E1032A4  ) 6
31038  7F23E5DDE10604C5  ) 6
31046  783DD5CD8578D1EB  ) 3
```

▶ ▶ ▶

MONITOR

```
31Ø54   DD7EØØD63ØFEØA38   ) 2
31Ø62   Ø2D6Ø7FEØØ28Ø419   ) Ø
31Ø7Ø   3D18FAEBC1DD231Ø   ) 8
31Ø78   DEEBDD21A27F3AA8   ) 3
31Ø86   7F32A47FC9Ø1ØEØØ   ) 5
31Ø94   11ØØ7ECD3C2ØCDAD   ) 8
311Ø2   78D83EØDD7C9Ø1ØE   ) 4
3111Ø   ØØ112C7ECD3C2ØCD   ) 4
31118   AD78D83EØDD7C9Ø1   ) 2
31126   ØCØØ113A7ECD3C2Ø   ) 5
31134   CDAD78D83EØDD7C9   ) 2
31142   Ø11ØØØ11ØE7ECD3C   ) 7
3115Ø   2ØCDAD78D83EØDD7   ) 6
31158   C9Ø1ØEØØ111E7ECD   ) 8
31166   3C2ØCDAD78D83EØD   ) 4
31174   D7C9ED73C67FED7B   ) Ø
31182   C27F2AC47FE5ED73   ) 6
3119Ø   C27FED7BC67FC9E6   ) 8
31198   385F3ACØ7F577BE6   ) 6
312Ø6   3Ø2822FE1Ø2815FE   ) 1
31214   2Ø28ØA7B17171717   ) 8
31222   AACB7F18177B1FAA   ) Ø
3123Ø   CB57181Ø7B1F1F1F   ) 6
31238   AACB4718Ø77B1717   ) 5
31246   17AACB7728Ø2A7C9   ) Ø
31254   37C9112F7FØ1Ø3ØØ   ) 8
31262   CD3C2ØC941424344   ) 7
3127Ø   45464A4C4D4E4F5Ø   ) Ø
31278   51525355565A716F   ) Ø
31286   DA6F1D7Ø7C7ØD872   ) 2
31294   FF7Ø357327717271   ) 1
313Ø2   BC71D6739671D975   ) 8
3131Ø   E371F2759F71DB76   ) 8
31318   E775FEF6EEE6DEDB   ) 5
31326   D6D3CEC63E38363Ø   ) 7
31334   2E28262Ø1E18161Ø   ) 8
31342   ØEØ6Ø11121222A31   ) Ø
3135Ø   323AC2C3C4CACCCD   ) 2
31358   D2D4DADCE2E4EAEC   ) 2
31366   F2F4FAFC7B736B63   ) 6
31374   5B534B4321222A36   ) Ø
31382   262E3435464E565E   ) 6
3139Ø   666E7Ø7172737475   ) 5
31398   777E868E969EA6AE   ) 4
314Ø6   B6BEFEF6EEE6DED6   ) 8
31414   CEC6BEB6AEA69E96   ) 6
31422   8E867E766E665E56   ) 6
3143Ø   4E463E2E261E16ØE   ) Ø
31438   Ø6FFFCFAF8F7F4F2   ) 7
31446   FØEFECEAE8E7E4E2   ) 5
31454   EØDFDCDAD8D7D4D2   ) 5
31462   DØCFCDCCCAC9C8C7   ) 7
3147Ø   C4C3C2CØ383Ø2Ø2Ø   ) 7
31478   181ØØØØØØØØØØØØØ   ) 2
31486   ØØØØØØØØØØØØØØØØ   ) Ø
31494   ØØØØØØØØØØØØØØØØ   ) Ø
315Ø2   ØØØØØØØØØØØØØØØØ   ) Ø
3151Ø   ØØØØØØØØØØØØØØØØ   ) Ø
31518   ØØØØØØØØØØØØØØØØ   ) Ø
31526   ØØØØ16ØØØC13Ø111   ) 5
31534   Ø31ØØ74D4F4E4954   ) 7
31542   4F521ØØØ11Ø713ØØ   ) 2
3155Ø   16Ø2Ø3416C746572   ) Ø
31558   2Ø6D656D6F727916   ) 2
31566   Ø3Ø3427265616B7Ø   ) 2
31574   6F696E7416Ø4Ø343   ) 6
31582   6F6E766572742Ø6E   ) 3
3159Ø   756D62657216Ø5Ø3   ) 3
31598   44656C657465 2Ø62   ) 6
316Ø6   7265616B7Ø6F696E   ) 6
31614   7416Ø6Ø34578616D   ) 7
31622   696E652Ø73746163   ) 6
3163Ø   6B16Ø7Ø346696C6C   ) 3
31638   2Ø6D656D6F72792Ø   ) 1
31646   626C6F636B16Ø8Ø3   ) 8
31654   4A756D7Ø2Ø746F2Ø   ) 2
31662   726F7574696E6516   ) 6
3167Ø   Ø9Ø34C6F6F6B2Ø61   ) 4
31678   742Ø627265616B7Ø   ) 8
31686   6F696E747316ØAØ3   ) 8
31694   4D6F76652Ø6D656D   ) 3
317Ø2   6F72792Ø626C6F63   ) Ø
3171Ø   6B16ØBØ34E756D62   ) Ø
31718   6572Ø2Ø6261736520   ) 3
31726   746F676C652Ø6E6F   ) 5
31734   7716ØCØ34F6E652D   ) 2
31742   737465 7Ø2Ø726F75   ) 1
3175Ø   74696E6516ØDØ35Ø   ) 1
31758   72696E746572 2Ø74   ) 3
31766   6F67676C652Ø6E6F   ) 2
31774   7716ØEØ351756974   ) 8
31782   16ØFØ35265676973   ) 8
3179Ø   746572 2Ø64697370   ) 1
31798   6C617916 1ØØ3537Ø   ) 6
318Ø6   656369667 92Ø656E   ) 4
31814   74727 92Ø76616C75   ) Ø
31822   657316 11Ø3557365   ) 2
3183Ø   2Ø6F746865722Ø7Ø   ) 6
31838   726F6772616D1612   ) Ø
31846   Ø356696577 2Ø6D65   ) 3
31854   6D6F72791613Ø35A   ) 7
31862   65726F2Ø72656769   ) 8
3187Ø   737465727316 14ØB   ) 7
31878   1ØØ176322E312Ø7F   ) 5
31886   2Ø4A6F6E2Ø456C6C   ) 6
31894   69732Ø3139383516   ) 4
319Ø2   15Ø513Ø11ØØ212Ø1   ) 3
3191Ø   417265 2Ø796F752Ø   ) 1
31918   7375726 52Ø3F2Ø2Ø   ) 8
31926   592F4E2913ØØ12ØØ   ) 1
31934   1ØØØ1615ØØ2Ø2Ø2Ø   ) 1
31942   2Ø2Ø2Ø2Ø2Ø2Ø2Ø2Ø   ) Ø
3195Ø   2Ø2Ø2Ø2Ø2Ø2Ø2Ø2Ø   ) Ø
31958   2Ø2Ø2Ø2Ø2Ø2Ø2Ø2Ø   ) Ø
31966   2Ø2Ø2Ø2Ø2Ø16ØØØØ   ) 1
31974   13Ø11ØØ15A383Ø2Ø   ) 2
31982   5265676973746572   ) 6
3199Ø   2Ø536574 16Ø2ØØ13   ) 6
31998   ØØ1ØØ24E6F726D61   ) 5
32ØØ6   6C2Ø526567697374   ) 1
32Ø14   6572733A16Ø4Ø41Ø   ) 8
32Ø22   ØØ412Ø2D16Ø41342   ) 4
32Ø3Ø   432Ø2D16Ø5Ø34445   ) 3
32Ø38   2Ø2D16Ø513484C2Ø   ) 2
32Ø46   2D16Ø7ØØ1ØØ4457 8   ) 1
32Ø54   6368616E6765 2Ø52   ) 6
32Ø62   6567697374657273   ) 8
32Ø7Ø   3A16Ø9Ø31ØØØ4127   ) 7
32Ø78   46272Ø2D16Ø91342   ) 7
32Ø86   2743272Ø2D16ØAØ3   ) 5
32Ø94   44274527 2Ø2D16ØA   ) 3
321Ø2   1348274C272Ø2D16   ) 4
3211Ø   ØCØØ1ØØ3537Ø6563   ) 3
32118   69616C2Ø52656769   ) 7
32126   73746572733A16ØE   ) Ø
32134   Ø31ØØØ4958 2Ø2D16   ) 1
32142   ØE134959 2Ø2D16ØF   ) 8
3215Ø   Ø353 5Ø2Ø2D16ØF13   ) 8
32158   5Ø432Ø2D161ØØ449   ) 3
32166   2Ø2D161Ø14522Ø2D   ) 6
32174   1612ØØ1ØØ1466C61   ) Ø
32182   67733A1612Ø7532Ø   ) 7
3219Ø   5A2Ø2D2Ø482Ø2D2Ø   ) 1
32198   5Ø2Ø4E2Ø431ØØØ16   ) 6
322Ø6   15ØØ1ØØ2496E7465   ) 4
32214   727275 7Ø74733A1Ø   ) 2
32222   ØØ16Ø41816Ø5Ø816   ) 2
3223Ø   Ø51816Ø9ØA16Ø91A   ) 5
32238   16ØAØA16ØA1A16ØE   ) 2
32246   Ø816ØE1816ØFØ816   ) 1
32254   ØF183A2Ø61646472   ) 2
32262   65737312Ø13E12ØØ   ) Ø
3227Ø   17Ø2ØØ3A2Ø6C656E   ) 8
32278   677468 12Ø13E12ØØ   ) 7
32286   17Ø2ØØ3A2Ø627974   ) 1
32294   6512Ø13E12ØØ17Ø2   ) 7
323Ø2   ØØ3A2Ø66726F6D12   ) Ø
3231Ø   Ø13E12ØØ17Ø2ØØ3A   ) 5
32318   2Ø746F12Ø13E12ØØ   ) 2
32326   16ØØØB1ØØ2557365   ) 2
32334   722Ø537461636B1Ø   ) 3
32342   ØØ16ØØØ91ØØ25573   ) 4
3235Ø   65722Ø7265676973   ) 5
32358   74657273 1ØØØØD61   ) 6
32366   292Ø2Ø2Ø2Ø2Ø433A   ) 3
32374   ØD62292Ø2Ø2Ø2Ø2Ø   ) 5
32382   423AØD63292Ø2Ø2Ø   ) Ø
3239Ø   2Ø2Ø453AØD64292Ø   ) 6
32398   2Ø2Ø2Ø2Ø443AØD65   ) 8
324Ø6   292Ø2Ø2Ø2Ø2Ø4C3A   ) 3
32414   ØD66292Ø2Ø2Ø2Ø2Ø   ) 4
32422   483AØD67292Ø2Ø2Ø   ) 4
3243Ø   2Ø46273AØD68292Ø   ) 7
32438   2Ø2Ø2Ø41273AØD69   ) Ø
32446   292Ø2Ø2Ø2Ø43273A   ) 8
32454   ØD6A292Ø2Ø2Ø2Ø42   ) 5
32462   273AØD6B292Ø2Ø2Ø   ) 5
3247Ø   2Ø45273AØD6C292Ø   ) 2
32478   2Ø2Ø2Ø44273AØD6D   ) 8
32486   292Ø2Ø2Ø2Ø4C273A   ) 8
32494   ØD6E292Ø2Ø2Ø2Ø48   ) 7
325Ø2   273AØD6F292Ø2Ø2Ø   ) 3
3251Ø   2Ø49583AØD7Ø292Ø   ) 1
32518   2Ø2Ø2Ø49593AØD71   ) 4
32526   292Ø2Ø2Ø2Ø2Ø413A   ) 7
32534   ØD72292Ø43617272   ) 5
32542   793AØD73292Ø2Ø5A   ) 7
3255Ø   65726F3A16Ø1ØØ15   ) 8
32558   Ø117Ø9ØØ1615ØØ1Ø   ) 6
32566   Ø1456E746572 2Ø76   ) 5
32574   616C75653A1ØØØØD   ) 1
32582   ØD17Ø4ØØ44656369   ) 3
3259Ø   6D616C3A2ØØD17Ø4   ) 3
32598   ØØ4865783A2ØØD17   ) 1
326Ø6   Ø4ØØ42696E617279   ) 1
32614   3A2Ø16Ø1ØA13Ø11Ø   ) 5
32622   Ø1427265616B7Ø6F   ) 6
3263Ø   696E747313ØØ1ØØØ   ) 8
32638   ØDØD17Ø7ØØ17Ø9ØØ   ) 4
32646   2E2E2E2E2ØØDØD13   ) 5
32654   Ø11ØØ212Ø1427265   ) 3
32662   616B13ØØ1ØØØ12ØØ   ) 7
3267Ø   2Ø6174 2ØØØØ5ØAØØ   ) 1
32678   ØØØØØØØØØØØØØØØØ   ) Ø
32686   ØØØØØØØØØØØØØØØØ   ) Ø
32694   ØØØØØØØØ5827ØØØØ   ) 8
327Ø2   3A5CØØØØFD7FØØØØ   ) 1
3271Ø   A46EØ1C3ØØØØØØØØ   ) 6
32718   ØØØØØØØØØØØØØØØØ   ) Ø
32726   C38E73ØØØØØØØØØØ   ) 5
32734   C39673ØØØØØØDØ5Ø   ) 2
32742   3ØØØCF5ØØCØ25CØE   ) 8
3275Ø   CØ5771ØEF3847317   ) 4
32758   C61EA761768C738E   ) Ø
32766   733EØØØØØØØØØØØØ   ) 5
```

Listing 3

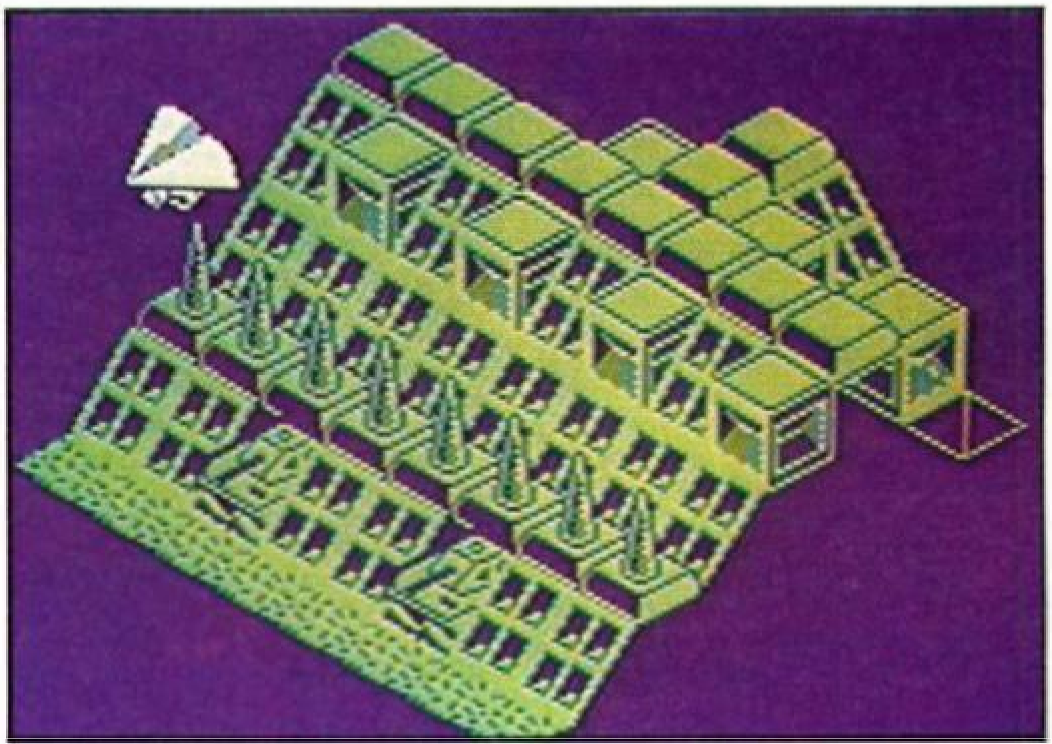
Error checker.

```
 1Ø REM ************************
 2Ø REM **    HEX  DUMPER     **
 3Ø REM ** by Jon Ellis 1985 **
 4Ø REM ************************
 6Ø FOR F=2835Ø TO 32767 STEP 8
 7Ø PRINT F;
 8Ø LET CHECK=Ø
 9Ø LET h$=""
1ØØ FOR g=1 TO 8
11Ø LET byte=PEEK (f-1+g)
12Ø LET hi=INT (byte/16)
13Ø LET lo=byte-16*hi
14Ø LET h$=h$+CHR$ (hi+48+(7 AN
D hi>9))+CHR$ (lo+48+(7 AND lo>9
))
15Ø LET check=check+byte*g
16Ø NEXT g
17Ø PRINT TAB 8;h$;
18Ø LET digit=check-9*INT (chec
k/9)
19Ø PRINT TAB 27;" ) ";digit
2ØØ NEXT f
21Ø STOP
```

GLIDER RIDER

Ride the thermals in Quicksilva's glide and destroy mission.

Glider Rider
Quicksilva
£8.95



This is a real dare devil mission. As Commander Glenn White you are dropped onto an island — your objective is to destroy the nuclear reactors that power the munitions plant owned by the Abraxas Corporation (Arms for Anyone, Anywhere, Any Reason).

For the mission you have been provided with a trail bike which converts into a hang glider. The trail bike will take you over rough territory and the conversion to hang glider is very slick — simply find a slope, career down it, reverse direction and the hang glider opens out like a butterfly and you are in the air. Staying in the air however is a different matter as practically all the installations on the island are heavily defended by ground to air lasers which if they lock onto you mercilessly sap your energy rating.

Puttering about on the motorbike is enjoyable and the grand tour of the island is recommended before attempting your objective. You can drive round the perimeter of the island without too much interference from laser outposts and appreciate a) the size of the island and b) the excellent and well thought out detail in the contoured 3-D graphic landscape. Then find an incline and take off. Mastering the controls of the hang glider takes some time as it seems to react as an actual hang glider would — bank too steeply and it stalls. If you are over the ocean and ditch you are immediately consumed by predatory sharks.

Flying around and across the island is fun but it's when the serious business of bombing hell out of things gets underway that a few flaws in the game are revealed. You can only bomb installations from the glider and are given a ration of nine (extra supplies can be picked up from ammo dumps by using the bike) but the question is what do you bomb and how do you know when you've hit it?

The instructions are hazy at best, hinting that 'power units are vulnerable' But what are power units? There are all kinds of structures on the island so where do you start? An index to buildings would have been a very useful aid.

The second problem is whether it's actually possible to destroy anything. Drop a bomb and there is no sign of an animated explosion.

One solution of course may be that I'm too jittery to control the hang glider or too myopic to aim properly but the fact is that I spent a great deal of time and concentration just attempting to register a hit on something — anything.

I tried dropping bombs from every conceivable height and angle on any potential target. The result? Nothing. (Bombs also have a disconcerting way of disappearing in mid air if you happen to switch screens as you fly over a target).

This, to say the least, is very annoying and it's all the more infuriating because everything else about the game is very appealing. Making things difficult for the player is all very well but getting a toehold on the game should be easier than this. This fact is the only thing that prevents Glider Rider from becoming a Monster Hit. It's a game I shall return to again in an effort to crack it, but I can't help thinking that this one should have been called Mission Impossible (literally) if someone hadn't thought of the title first.







THE BEST OF BEYOND

Christmas is coming and Beyond are ready with an early prezzie for strategy and adventure fans.

The Best of Beyond
Beyond Software
£9.95

Beyond have always been stronger on strategy/adventure than on arcade games (do I hear Superman being mentioned anywhere?), and now, just to tide us all over while we're waiting for Dark Sceptre and Star Trek to arrive, they've gathered together four of their best-known titles together onto a compilation tape.

The games assembled here have all had their fair share of acclaim — **Doomdark's Revenge, Shadowfire** and **Enigma Force,** with only **Sorderon's Shadow** as a slightly less well-known game.

It seems almost pointless to rehash Doomdark's Revenge, especially as Mike Singleton's 'landscaping' technique caused such a fuss when it first appeared in Lords of Midnight. But Doomdark took the technique even further, to produce a massive fantasy epic in which, playing the part of Prince Luxor and controlling various other characters and armies, you embark on a quest to free the land of Midnight from the threat of Shareth the Heartstealer. The game allows you to enter commands via single key entry (w=look, q=move), and the six thousand locations, 48,000 landscaped views and the various characters and objects that you have to deal with make for a vast and complex game that has been keeping hardened adventure addicts glued to their keyboards ever since it was first released.

A second tape in the package also includes a recording of a dramatised version of the story of Doomdark's Revenge — a must for lovers of purple prose.

The Shadow

Sorderon's Shadow is next on side one, and again there's a touch of the landscapes as you attempt to free the land of Elindor from the rule of the not-very-nice wizard, Sorderon.

If you can complete the nine tasks set by The Un-named One you may be able to cast the final spell that will end Sorderon's power. To complete these tas' you'll have to trek around El:ndor collecting various magical objects from the characters who inhabit the land. The game is more of a conventional adventure than Doomdark, despite the landscaped graphics, and entry of commands is of the normal 'Get the sword, kill the oik' method rather than single key entry (though the command editor is fairly sophisticated). Sorderon's Shadow suffers a little bit because the landscaped graphics invite a not really appropriate comparison with Doomdark, possibly leading you to expect more than the game delivers. Underneath the visually attractive presentation, Sorderon's Shadow is a fairly good, but by no means outstanding adventure.

The Enigma team

Shadowfire and Enigmaforce (both written by Denton Designs) are sensibly put together on side two of the tape, as they both feature the adventures of the Enigma team. Shadowfire caused a huge fuss when first released as it was one of the first games to feature a truly sophisticated icon control system. Both games put you in control of Zark, Syylk, Sevrina, and Maul of the Enigma team

Doomdark's Revenge



Enigma Force



as they attempt to foil the plans of the galactic despot, General Zoff.

The challenge presented by the games lies in balancing the strengths and weaknesses of your team members and deploying them effectively against the forces controlled by Zoff. As with Doomdark, these games do not offer a single, clear cut path to victory and there's always room for improvement and new tactics. Shadowfire featured only the icon system and a static display of the characters and their positions, but in Enigma Force this had been upgraded to an animated display in which you could actually see the characters move in response to your commands.

My only quibble about these games was that I found the icon system a bit fiddly to cope with, but there's no denying the difficulty of the challenge that the two games present you with or the quality of the games' design. With four games like this all on the same tape Beyond have produced a high-class compilation which, assuming you haven't already got most of the games, represents good value for followers of adventure and stategy games.

MONSTER HIT

QL COLUMN

Anthony Stewart faces the inevitable.

THE PCW Show has come and gone for yet another year and there was hardly a QL in sight. Digital Precision was an exception and the company had the courage to flaunt its sentiments with a sign reading: **"QL only and proud of it!"** But everything else was Amstrad or Atari and even the new Spectrum +2 seemed to take a back seat to Amstrad's launch of an impressive-looking, inexpensive PC obviously aimed at capturing the poor-man's-IBM market. Sadly, the QL looks as dead as the proverbial door nail but let's postpone the obituary for a little while and talk about happier things for a bit.

THOR, CST's reincarnated, higher life-form, QL was there in force and it's a good-looking machine. CST and I have something in common; neither one of us can manage to get any comment out of Amstrad! While I'm still waiting for that elusive Amstrad PR man to "check with the client and get back to me", CST has been taking the bull by the horns and telexing the company directly to try and reach an accommodation over Thor's use of QL circuit boards. They haven't heard a ticky-boo so maybe it's not just me, maybe Amstrad doesn't talk to anybody about anything. I did have to grin, however, at the sight of a CST exec (company ID badges conspicuous by their removal) taking splashy hand-outs from a smiling, PC-promoting Alan Sugar. It's the sort of thing that makes you want to root for the underdog even if you aren't willing to put a few quid on him.

QL mouse

Those hard-to-find West Germans finally reappeared at the Show and I got a chance to play with their QL mouse and software (a desk-top manager and a Superbasic extension) package that sells for around £90. It seems excellent in all respects and makes an ideal alternative to Eidersoft's ICE and mouse combination (which costs about the same) but — save for the odd computer show — you can only reach these people by contacting them at Giga-soft in Dusseldorf (full address in the odd ZX Computing advert or supplied upon request) so Eidersoft probably doesn't have a lot to worry about.

There was no new QL software worth mentioning and, more surprisingly, very little new stuff for the Spectrum. So as not to totally waste my tube-fare, I got a couple of games for my Spectrum. In particular, Trivial Pursuit (from Domark) looks a winner and, for the less trivially-minded, action-games players amongst you, Strike Force Cobra (Piranha), Night Rider (Ocean) and TT Racer (Digital Integration) are providing my nine-year-old son (and his father) with endless hours of amusement. In case you suspect that his father is trying to avoid talking about the inevitable, you happen to be just about spot-on and — here it is — the QL is now history in the past tense. The quantum leap is dead and gone.

R.I.P.

I talked to quite a few of the major QL suppliers and software houses and, when I could get them to stop going on about the new Amstrad PC, they all said much the same thing. It's finished and Amstrad rules the day. Microdeal, which has supplied numerous QL games and an excellent flight simulator, told me (on their stall in the Atari exhibition) that nobody's really sending in any more QL software for possible publication. Obviously our games-designing friends throughout the land, having spotted the inevitable, have moved on to fresher and more profitable climates. Some of the other, more heavy-duty software houses said much the same thing after all the jargon was translated into day-to-day English. Alright, so there's Thor but it's a different type machine at a price few home-micro users are going to look at for long. As a business package or system for the richer home user, it looks very good but it's not the quantum leap that Sir Clive tried to give us at every-day prices.

So don't expect much in the way of new QL software. There will be some in the days to come but the flood gates that only started to open a few short months before Sinclair computers got themselves a new owner have shut again and we are back into a square-one-situation (or worse). But all isn't gloom and despair. There's a large amount of quality software for the QL already on the market to keep most owners in business for the lifespan of their computers. Those of you who are toying with the idea of buying one ought to keep thinking and not look elsewhere too quickly. If you want a good computer at a rock-bottom price (which just might get a bit more rock-bottom before too long) which will last for a few years and which has a lot of very good software (which, if not exactly selling for next to nothing is economically priced and is definitely going to get cheaper), a QL is still something worth looking at.

The QL's ghost, like old John Brown's, goes marching on even if the body lies a-moulding in the ground. Go have a look at the new Amstrad PC but forget about all the flashy sales stuff telling us that it's the greatest thing since pop-top beer cans and fated to make IBM decline into obscurity. What you are going to see is the gadget that Sir Clive tried to market but didn't. It's got the built-in disc drive that the QL should have had instead of those temperamental microdrives. It has an excellent keyboard and comes with a monitor. The basic package costs some £399 and a few hundred more will give its owner a top-notch system. Amstrad's only mistake as far as I can see is to endlessly go on about the PC's IBM compatability and why it's a better deal. Both are no-doubt true, but who really wants to buy an imitation IBM when they might just be able to save enough coppers to buy a real one?

All this is what Sir Clive tried to do a few short years ago (except that he had the guts if not the good sense not to go for an IBM clone) and it's just that Amstrad is profitting from his errors and their own undoubted marketing expertise. Amstrad is even promising us a high-quality chess package for the PC but what did Sinclair give us straight-away (when most software houses were still waiting to see how things were going to shape up)? A first-class chess package that's still unbeatable. When you do have a look at what will undoubtedly be a disgustingly successful business venture, remember the famous words of what's-his-name:

"Of all the words of paper of pen, the saddest are these. 'It might have been'." I can assure you that Sir Clive will.

SPECTRUM/QL LINK-UP

They said it couldn't be done, but David Nowotnik shows how to get a Spectrum and QL communicating with each other.

Last month I described how data could be passed between the Spectrum and QL computers via their network ports. We discovered that while communication by this route is possible, certain incompatibilities between the two computers' network systems made data transmission and reception slow and unreliable. But both computers offer an alternative — the RS232 ports, and this, you should find, offers far greater reliability and speed.

As with network communication, you will need an Interface 1 for your Spectrum to allow the QL and Spectrum to 'talk' to each other. In addition, you will need a special lead. If you are handy with a soldering iron, then you might like to try making up your own lead. If not, then try one of the companies who regularly advertise in 'ZX Computing' and other magazines with special QL RS232 leads.

On both computers the connectors are non-standard for RS232 lines. At the QL end the connector is a British Telecom plug (the sort you'll have at the wall end of your telephone if your telephone is transferrable). And for Spectrum connections you'll need a nine pin D-plug; the sort that will fit into Atari joystick plugs. Both types of plug, and the lead to connect them can be obtained from specialist electronic and telephone supply shops.

The BT plug requires great skill (and a specialist tool) to wire up the five leads necessary to complete the link. If you don't have the confidence to try this for yourself, you could 'sacrifice' the RS232 lead supplied to link the QL to a printer, by removing the standard RS232 plug and connecting the D plug.

Pin connections to be made at both the QL and Spectrum ends are listed in table 1. The QL has two serial ports, and both are wired slightly differently, so decide in advance which serial port you wish to use (the examples used later on use ser1), and wire the D plug according to the appropriate column in the table. If you elect to sacrifice a QL printer cable, then the appropriate wire colours are shown alongside the D-pin number to which it is connected. D-pin numbers are shown in Fig.1. (viewing the plug from the back; the side to which leads are soldered). If you are wiring a BT plug, then pin 1 is the one farthest from the plastic leg of the plug.

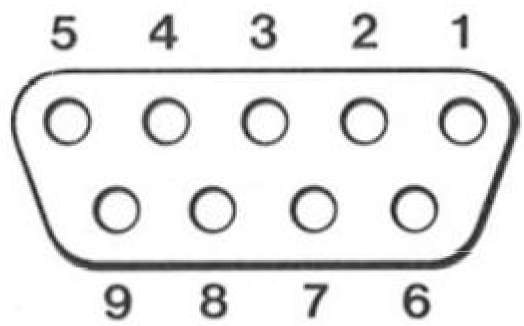


Fig.1. Diagram of D plug pin connections

Getting wired

Of the five wires which connect the two serial ports, two are for data communication (data flow is unidirectional on each wire), and two provide simple switches to permit transmitter and receiver to tell each other if they are ready for data communication. These work by placing a voltage on the line to indicate a state of readiness. That voltage is relative to 'ground', which is the fifth line to connect both computers.

Once you have connected or purchased your RS232 lead you might still have problems getting the two computers talking to each other. This is because the RS232 'standard' is far from being absolutely standard. With RS232 data is transmitted serially, i.e. one bit after another along a single line. This data can travel at different speeds, and transmitter and receiver must be set to talk and listen at the same speed. The speed of transfer is known as the baud rate (bits per second), and we'll see in a moment how the baud rate is set on both computers.

There are other possible RS232 mismatches between transmitter and receiver which can cause problems; things called parity bits and handshaking. Fortunately, the default setting on QL and Spectrum are similar, so they need not concern us here. But beware of these potential problems when linking your computer to any other device.

So, with your computer turned on, and RS232 lead connected, you are ready to test out your wiring efforts. Try the test routine in Fig.2. Type is the program lines to each computer, **SAVE** them, and **RUN** both computers. Once both programs are running, you need only type on the QL keyboard. The INPUT cursor will flash in window #0, and anything you type in and ENTER will (if all is well) appear on the Spectrum screen, then on the QL screen. The word or message has to be successfully transmitted to the Spectrum, then transmitted back to the QL before it appears on the QL screen. So, your message on the QL screen means that communication is working in both directions. Note from the listing that it is only necessary to open a single channel on both QL and Spectrum.

Table 1. RS232 Lead – wiring connections

D pin (connect to ser1)	BT Socket	D pin (connect to ser2)
7 black	1	7 black
3 white	2	2 white
2 green	3	3 green
5 blue	4	4 blue
4 red	5	5 red

Communication on a single RS232 channel is bi-directional.

The next step is to see which of the various BAUD rates is most reliable. The QL and Spectrum have six transfer rates in common. The listing in Fig.3 is designed to test them out. Again type in the listings, SAVE and RUN them.

In this test, the Spectrum is again the 'slave' of the QL, receiving messages, printing them on the screen, then transmitting them back to the QL, where they are tested for errors. The message transmitted appears in the DATA lines 60-90 of the QL listing. Also transmitted are a set of six control codes. These are code numbers which do not correspond to any of the character codes shared by the two machines. The Spectrum RS232 channel has been opened to type "b" (see your Interface 1 handbook) in line 40 to permit these characters to be transferred. In any Spectrum/QL RS232 communications, it is also preferable to use the "b" (binary) rather than "t" (text) option on the Spectrum side, otherwise odd results might occur.

So, the program loops round from 9600 Baud to 300 baud (line 150 in the QL listing, 40 in the Spectrum listing) sending the same message and set of numbers from QL to Spectrum, then back again. The result of every test is printed on both screens, with the number of errors, and time taken given on the QL screen. The test of time is carried out using the QL system variable at address 163886. Set to zero (line 160), it increments fifty times a second, so when it is PEEKed (line 290) and divided by 50, it gives the time taken more accurately than using DATE (which only gives the time to the second). After each test the program halts, waiting for you to press any key on the QL keyboard.

```
Fig.2. Simple communication test routine

a) Spectrum listing

10 REM    Test of RS232
20 REM
30 FORMAT "b";1200
40 OPEN #4;"b"
50 REM
60 INPUT #4;a$
70 PRINT a$
80 PRINT #4;a$;CHR$ 10;
90 IF a$="stop" THEN STOP
100 GOTO 50

b) QL Listing

10 REMark  Test of RS232
20 REMark
30 BAUD 1200
40 OPEN #4, ser1
50 REPeat loop
60    INPUT #0,a$
70    PRINT #4,a$; CHR$(13);
80    INPUT #4,b$: PRINT b$
90    IF b$="stop" THEN EXIT loop
100 END REPeat loop
```

Time

There is a fixed time on each test devoted to the testing and printing of results, so the influence of different baud rates can only be judged by the differences in time taken. You should find that the accuracy of transfer should not vary much with speed, and very few faults should occur. Surprising, the only fault I found was at the lowest transfer rate, 300 baud, when data transfer seemed to fail altogether. Try the test a few times for yourself, and confirm these results. It will mean that there is no disadvantage (in terms of reliability), and much to be saved in time by always selecting the maximum baud rate of 9600, which happens to be the default value on the QL.

Last month I gave you a program to transfer BASIC programs from Spectrum to QL using the network system. This

```
Fig.3 Test of speed and reliability

a) Spectrum Listing

10 REM Reliability Test
20 REM
30 LET x=19200: LET h$= CHR$ 10
40 LET x=x/2: FORMAT "b";x
50 OPEN # 4;"b": CLS
60 FOR j=1 TO 4
70 INPUT #4;a$
80 PRINT a$
90 PRINT #4;a$;h$;
100 NEXT j
110 INPUT #4;a$
120 PRINT #4;a$;h$;
130 PRINT
140 FOR i=1 TO 6
150 PRINT  CODE a$(i)
160 NEXT i
170 PRINT "Baud =";x
180 INPUT #4;a$
190 CLOSE # 4
200 IF x>300 THEN  GO TO 40

b) QL Listing

10 REMark     RS232 test progran
20 REMark
30 x=19200: e$="": h$= CHR$(13)
40 DIM a$(4,28): RESTORE
50 FOR i=1 TO 4: READ a$(i): END FOR i
60 DATA "This is a test of speed and "
70 DATA "reliability of the transfer "
80 DATA "of data between the QL and  "
90 DATA "Spectrum computers."
100 DATA 120,254,255,0,23,5
110 FOR i=1 TO 6
120    READ z:e$=e$& CHR$ (z)
130 END FOR i
140 REPeat loop
150    x=x/2: BAUD x: er=0: CLS
160    POKE_W 163886,0
170    OPEN #4, ser1
180    FOR j=1 TO 4
190      PRINT #4;a$(j);h$;
200      INPUT #4,v$: PRINT v$
210      FOR k=1 TO LEN (v$)
220        IF v$(k)<>a$(j,k) THEN er=er+1
230      END FOR k
240    END FOR j
250    PRINT #4;e$;h$;
260    INPUT #4;v$
270    IF v$<>e$ THEN er=er+1
280    PRINT\ "Baud = ";x;" Errors = ";er
290    t=INT (PEEK_W (163886)/50)/10
300    PRINT "Time = ";t;" seconds"
310    s$=INKEY$(-1): v$="next": PRINT #4;v$;h$;
320    PAUSE 50: CLOSE #4
330    IF x=300 THEN EXIT loop
340 END REPeat loop
```

Fig.4. RS232 Transfer of Spectrum programs to QL

a) Spectrum Listing

```
9980 FORMAT "b";9600
9981 GO SUB 9997
9982 LET k=y-x: LET r=0
9983 OPEN # 5;"b": PRINT #5; STR$ k; CHR$ 10;
9984 FOR i=1 TO k
9985 GO SUB 9997
9986 LET z$= CHR$ ( PEEK (x+r))
9987 LET r=r+1: PRINT #5;z$;
9988 NEXT i
9989 CLOSE # 5
9990 STOP
9997 LET x= PEEK 23635+256* PEEK 23636
9998 LET y= PEEK 23627+256* PEEK 23628
9999 RETURN
```

b) QL Listing

```
100 REMark Import Spectrum Programs
110 REMark via the RS232 line
120 :
130 REMark Initialise
140 :
150 CLS :PRINT " Initialising - please wait"
160 a= RESPR (0)
170 IF a> 221184 THEN a= RESPR (a-221184)
180 NET 1
190 z$= CHR$(13)
200 position=a: RESTORE 1000
210 DIM keyword$ (91,10), key_len (91)
220 FOR i=1 TO 91
230   READ a$: keyword$ (i)=a$
240   key_len (i)= LEN (a$)
250 END FOR i
260 :
270 REMark Transfer program from Spectrum
280 :
290 BAUD 9600: OPEN #5, ser1
300 INPUT #5;num$: num=num$
320 PRINT \" Receiving the program"\" from the Spectrum"
330 FOR i=1 TO num
350   x$= INKEY$(#5,-1)
360   POKE position, CODE (x$)
370   position=position+1
400 END FOR i
430 CLOSE #5
440 PRINT \\" Transfer complete"\\
450 :
600 REMark Create file of BASIC lines
610 :
620 PRINT " Place a formatted cartridge"
630 PRINT " in mdv2_, then press any key"\\
640 x$= INKEY$(-1)
650 DELETE mdv2_temp_bas
660 OPEN_NEW #4,mdv2_temp_bas
670 REPeat loop2
680   REMark line number first
690   a$=256*PEEK(a)+PEEK (a+1): a=a+4
700   IF a$="9980" THEN EXIT loop2
710   REPeat loop3
720     z= PEEK (a)
730     SELect ON z
740     =32 TO 127: a$=a$& CHR$(z)
750     =128 TO 143: a$=a$&'()'
760     =144 TO 164: a$=a$& '('&CHR$(z-79)&')'
770     =165 TO 255
780       x$=keyword$ (z-164): x$=x$(1 TO key_len (z-164))
790       a$=a$&x$
800     =13: EXIT loop3
810     =REMAINDER : a=a+5
820     END SELect
830     a=a+1
840   END REPeat loop3
850   PRINT #4; a$
860   PRINT a$
870   a=a+1: IF a>=position THEN EXIT loop2
880 END REPeat loop2
890 CLOSE #4
1000 DATA ' RND ',' INKEY$ ',' PI ',' FN '
1010 DATA ' POINT ',' SCREEN$ ',' ATTR ',' AT '
1020 DATA ' TAB ',' VAL$ ',' CODE ',' VAL '
1030 DATA ' LEN ',' SIN ',' COS ',' TAN '
1040 DATA ' ASN ',' ACS ',' ATN ',' LN '
1050 DATA ' EXP ',' INT ',' SQR ',' SGN '
1060 DATA ' ABS ',' PEEK ',' IN ',' USR '
1070 DATA ' STR$ ',' CHR$ ',' NOT ',' BIN '
1080 DATA ' OR ',' AND ',' <= ',' >= '
1090 DATA ' <> ',' LINE ',' THEN ',' TO '
1100 DATA ' STEP ',' DEF FN ',' CAT ',' FORMAT '
1110 DATA ' MOVE ',' ERASE ',' OPEN # ',' CLOSE # '
1120 DATA ' MERGE ',' VERIFY ',' BEEP ',' CIRCLE '
1130 DATA ' INK ',' PAPER ',' FLASH ',' BRIGHT '
1140 DATA ' INVERSE ',' OVER ',' OUT ',' LPRINT '
1150 DATA ' LLIST ',' STOP ',' READ ',' DATA '
1160 DATA ' RESTORE ',' NEW ',' BORDER ',' CONTINUE '
1170 DATA ' DIM ',' REM ',' FOR ',' GO TO '
1180 DATA ' GO SUB ',' INPUT ',' LOAD ',' LIST '
1190 DATA ' LET ',' PAUSE ',' NEXT ',' POKE '
1200 DATA ' PRINT ',' PLOT ',' RUN ',' SAVE '
1210 DATA ' RANDOMIZE ',' IF ',' CLS ',' DRAW '
1220 DATA ' CLEAR ',' RETURN ',' COPY '
```

was reliable only when each byte value was transmitted as a number, and the channel opened and closed with each transmitted number. As you have seen, the RS232 line is far more reliable, allowing data to be sent as a stream of 8-bit characters at 9600 bits per second. To make use of this faster and more reliable transfer rate (particularly useful in longer programs) program transfer can be improved using the RS232 line. Fig.3. contains the listing of the transfer program converted for use with the RS232 link. To use the programs, **MERGE** the Spectrum listing with the program you wish to transfer, and **RUN** the QL program, with a microdrive cartridge in mdv.2. Type and enter **RUN 9980** on the Spectrum. The program will be transferred, then translated to an ASCII file at the QL end, and stored on microdrive under the file name of 'temp_bas'. This can be loaded into the QL as a normal BASIC program. The QL adds the word 'Mistake' to the beginning of any line which is syntactically incorrect, and you may well see several of these when you list the program. In addition to these modifications, other changes will no doubt be necessary for the program to run successfully on the QL. But transfer of the program can save a lot of effort compared with re-typing the whole program.

Another benefit QL users could derive is in the transfer of the Spectrum screen display to the QL. Despite the limitations on its screen display, some excellent colour pictures can create for the Spectrum. With comparatively little available for the QL in this respect, programmers could transfer their favourite Spectrum screen displays, then import them into a graphics program (e.g. GraphiQL) for modification. The listing in Fig.5. allows the screen display to be transferred.

Screens

Once both listings are entered and saved, RUN them. The Spectrum screen asks you to press play on your tape recorder. It assumes that you have a tape in the recorder containing the screen display you wish to transfer, and that the screen display file is the first file it will find on the tape. In the meantime, the QL screen is asking for a file name for the screen. Once the transfer is complete, the screen display is saved to microdrive 2 using that filename.

Once the display is loaded on the Spectrum screen, the border will flash, indicating transfer of data via the RS232 line. Attribute data is transferred first, and saved in an array (att_array$) in the QL. At the same time, the paper attribute colours are displayed on the QL screen. In order to achieve the spectacular displays on the Spectrum's limited display file, some clever tricks have been used, and PAPER and INK are not always what you think they are. By first printing all of the PAPER blocks, you can see the clever way Spectrum pictures are made up. Next all the pixels are transferred. This is a slow business; it's not the transfer rate which is a problem, but all the calculations which are necessary by both computers. The Spectrum has to calculate the byte address of each block of eight pixels. This it does in line 260 with the aid of a small machine code routine installed in the user defined graphics area of RAM (installed using the subroutine starting at line 1000 of the Spectrum lising). the QL works out which bits in each byte are INK, and which are PAPER, and laces the appropriate pixel colour on the screen. Unless you are fascinated by the picture slowly building up on the QL screen, go and enjoy a cup of tea while this is going on! The display will fill a QL monitor screen, so you may find it will overflow if your have a TV as your QL screen.

There can be many applications for the transfer of data between two computers. In

this article, and the one last month, you should see how this is possible (knowing the limitations) between Spectrum and QL, and the principles developed could be applied to any application you may require.

Fig 5. Transfer of Spectrum screen display to QL

a) Spectrum listing

```
10 GO SUB 1000
20 PRINT "Press Play on your tape recorder"
30 LOAD "" SCREEN$
50 OPEN # 4;"b";9600
120 FOR i=0 TO 31
130 FOR j=0 TO 23
140 PRINT #4; CHR$  ATTR (j,i);
150 NEXT j
160 NEXT i
170 PRINT #4;"pixels";
230 FOR i=0 TO 192
240 FOR j=0 TO 31
250 POKE 32747,i: POKE 32755,i: POKE 32763,i
260 LET bt=16384+32* USR 32760+256* USR 32744+4* USR 32752+j
270 LET p= PEEK bt
280 PRINT #4; CHR$ p;
290 NEXT j
300 NEXT i
310 CLOSE # 4
990 STOP
1000 RESTORE
1010 FOR i=32744 TO 32767
1020 READ a: POKE i,a
1030 NEXT i
1040 RETURN
1050 DATA 6,0,62,0,230,7,79,201
1060 DATA 6,0,62,0,230,56,79,201
1070 DATA 6,0,62,0,230,192,79,201
```

b) QL Listing

```
10 REMark Spectrum Screen Transfer
20 :
30 INIT
40 INPUT \\"File name for screen display ";name$
50 PAPER_ATT
60 delay: PAINT_INK
70 SBYTES "mdv2_"&name$,131072,32768
80 STOP
90 :
100 DEFine PROCedure INIT
110 WINDOW 512,256,0,0: PAPER 0
120 MODE 8: CLS
130 DIM att_array$(32,24)
140 BAUD 9600
150 OPEN #4, ser1
160 END DEFine INIT
170 :
180 DEFine PROCedure BBlock (i,x,y)
190 BLOCK 16,10,16*x,10*y,i
200 END DEFine
210 :
220 DEFine PROCedure PAPER_ATT
230 LOCal i,j,a$,att
240 FOR i=0 TO 31
250   FOR j=0 TO 23
260   a$=INKEY$(#4,-1): att=CODE(a$)
270   att_array$(i+1,j+1)=a$
280   att=(att&&56)/8: BBlock att,i,j
290   END FOR j
300 END FOR i
310 END DEFine PAPER_ATT
320 :
330 DEFine PROCedure PAINT_INK
340 LOCal z,z$,ok,i,j,k,row
350 row=0
360 FOR i=0 TO 191
370   FOR j=0 TO 31
380     z$=INKEY$(#4,-1)
390     att=CODE (att_array$(j+1,(i DIV 8)+1))
400     ik=att&&7: pap=(att&&56)/8
410     z=CODE (z$)
420     FOR k=7 TO 0 STEP -1
430     IF (z MOD 2) THEN
440       BLOCK 2,1+((i MOD 3)=0),2*(j*8+k),row,ik
450       ELSE
460       BLOCK 2,1+((i MOD 3)=0),2*(j*8+k),row,pap
470       END IF
480       z=z DIV 2
490     END FOR k
500   END FOR j
510   row=row+1+(i MOD 3=0)
520 END FOR i
530 END DEFine PAINT_INK
540 :
550 DEFine PROCedure delay
560 LOCal x$,count,c$
570 x$="pixels": count=0
580 REPeat loop1
590   c$=INKEY$(#4,-1)
600   IF c$=x$(count+1) THEN
610   count=count+1
620   ELSE : count=0
630    END IF
640    IF count=5 THEN EXIT loop1
650 END REPeat loop1
660 c$=INKEY$(#4,-1)
670 END DEFine delay
```

Shock horror! A new QL game from Rubicon Systems.

Dragonhold

Dragonhold
Rubicon Computer Systems
£19.95

In the early days of computer programs there appeared several rather simple attempts at role playing D&D type games. None provided the sophistication and variety of action of the original game but they had charm and varying degrees of success. These developed in many ways into the games we now know and love on most machines. The QL however, never attracted the same degree of enthusiasm and ingenuity that programmers lavished on other machines.

Dragonhold has many features in common with these early games, but I must add that they have tried quite successfully to combine adventure and animation and strategy. The problem is that fans of each specific genre may find this hybrid unsatisfying, I just hope that there are enough 'all rounders' to buy, enjoy and make this a success.

The game is a traditional D&D type, you begin in the Dark Caverns and have to search for the Elixir of Life. There are six different areas and a host of creatures which react to you in different ways. You can attack or even talk to them if they are friendly.

Movement is achieved by cursor keys or joystick and is depicted graphically in the central window. Other windows provide information on status, text input, and messages, the graphics are mode 4 graphics and though not stunning, are certainly adequate for the game.

My greatest grievance against most arcade adventures is that there is usually no time to just sit and think. A Freeze option is supplied to allow this along with a full complement of save and load game options. The adventure elements are catered for in the most comprehensive system I have seen for this type of game, letter keys A to X have associated VERB or NOUN words to the extent of 11 verbs and 24 nouns. For instance, key 0 doubles as OPEN and OPAL. It means that adventure purists will find the joy of discovering the right words and the surprise of finding unknown objects has been eliminated. However when playing the game it adds greatly to the range of actions you can perform compared with a purely graphical program.

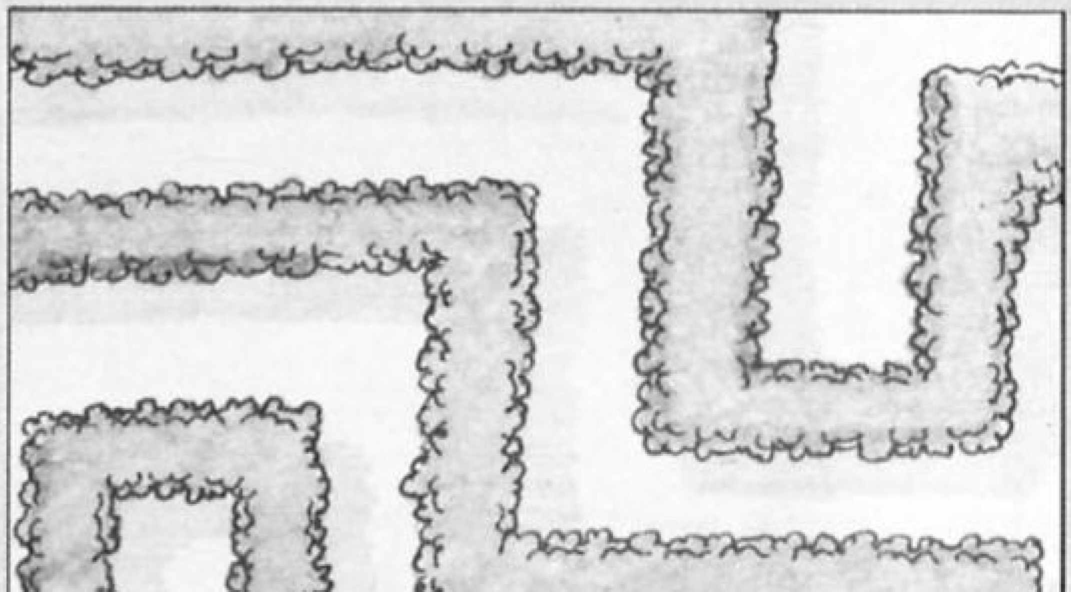
The ability to fight, converse and barter are all included and the options are very comprehensive. I found that its very lack of clever techniques, no icons, no multi-key combination presses, added to its appeal and soon I had become completely absorbed in the problems, not mind bending, but enough to make you pause for thought.

I liked it, a pleasant change and surprisingly addictive.



QL ERROR TRAPPING

Mel Mackaron presents an invaluable aid for QL programmers — a simple and effective error trapping technique.



The Sinclair QL is a wonderful computer, and SuperBASIC is a very good programming language, but, for the person trying to use his QL to write a fairly professional application, there has, until now, been a significant stumbling block: the 'absence' of an error-trapping command. Most micro-computers have a keyword combination such as ON ERR — GOTO xxx to help with controlling end-user errors or I/O errors. But nowhere in the Sinclair documentation, including the QL User Guide and the QL Technical Guide is there any mention of such an aid.

Ordinarily, the lack of such a command would not be a problem. If, in a utility you have written for yourself, an error occurs, you should know your program well enough to re-enter at a suitable point. But what about another user who does not have your knowledge of the program's structure? If he makes an error causing the program to terminate, he may not know how to get back in.

Rom Secrets

If you have faced this dilemma, you may have felt that you only had limited options: you could type a message to the user such as 'In case of an error, type "GOTO xxx" to re-enter the program'; you could attempt to learn assembly language well enough to write you own error-retrieval routines; you could give up. However, you do have a fourth option. Hidden in the QL's ROM (and not documented in the above-mentioned books) is the keyword combination **WHEN ERRor — RETRY — END WHEN.**

The concept behind **WHEN ERRor** is fundamentally the same as an ON ERR — GOTO statement; however, the syntax and coding are somewhat different. Nevertheless, once you

know how to use this tool, you will be able to give your program a more professional appearance.

When beginning work with **WHEN ERRor**, you must remember a couple of rules. First, do not try to exit from it by using a **GOTO**. This may cause unpredictable results later on, such as a message like 'error during WHEN processing'. Second, the **WHEN ERRor** routine needs to precede the point at which the error could occur. Also, if you are using separate **WHEN ERRor** routines, remember that the latest one takes precedence. Therefore, you must always keep in mind the user's location in the program so that the correct errors will be trapped.

The four sample program segments I have included show different levels of error-trapping. In the fact, all we shall rely upon is that which is built into the computer. In the second, the programmer is protecting against anything except an I/O error. The third demonstrates a basic introduction to the **WHEN ERRor** construct. The fourth shows how you can trap for multiple errors within one routine. Within these segments, I have also tried to show a bit of what is generally termed 'accepted programming practice', something which may help you to improve your own programming.

Deliberate mistakes

All program segments rely upon two type of errors. First, we ask for a number to be entered. On the QL, if a string (such as 'hello') were entered instead of a number, you would get error -17 (error in expression). Then, we summon a directory of MDV2. If there were no cartridge in that drive, we would get error -7 (file or device not found). You should enter all listings (one at a time) and then RUN them. When you do run them, purposely make the errors you are testing for. In that way, you will learn how each trapping method works and how we are improving on the basic, machine-level trapping.

Listing One works only with machine-level control. In other words, when the computer encounters an error, program execution stops with an erro message. You may then RUN the program again. While this method of detecting faults may be informaative, it is also fairly primitive and frustrating, especially for the user who is not well-versed in the operation of computers.

Listing Two show what a reasonably competent programmer might do for error control. In this one, we are

Listing One

```
 10 REMark This program demonstrates programming with no
 20 REMark at error-trapping.  It relies solely on machine
 30 REMark level error reports.
 40 REMark
 50 MODE 0 : PAPER 0 : INK 7 : CLS : REMark Set Display
 60 INPUT 'Enter a Number:  ';num
 70 PRINT \'Your number was ' & num
 80 PRINT \'Press any key to continue...'
 90 PAUSE : REMark waits for a keypress
100 CLS
110 DIR MDV2_
120 PRINT \'Program successfully completed.'
130 STOP
```

Listing Two

```
 10 REMark This section demonstrates error-trapping efforts by
 20 REMark the programmer who does not use WHEN ERRor.  It
 30 REMark checks for correct user input, but cannot control
 40 REMark I/O errors efficiently.
 50 REMark
 60 MODE 0 : PAPER 0 : INK 7 : REMark Set Display
 70 CLS : INPUT 'Enter a Number:  ';num$
 80 REMark Input is as a string which coercion will convert to
 90 REMark a number.
100 IF num$ = '' : GO TO 70 : REMark User only pressed ENTER
110 IF num$(1) < '0' OR num$(1) > '9' : GO TO 70
120 REMark User did not enter a number
130 num = num$ : REMark Correct entry, so convert it
140 PRINT \'Your number was ' & num
150 carry_on
160 CLS
170 PRINT 'Following is a directory of MDV2'
180 PRINT \'If you get an error, type "GOTO 180"'
190 carry_on
200 DIR MDV2_
210 PRINT \\'Program successfully completed!'
220 STOP
230 DEFine PROCedure carry_on
240   REMark merely a routine to wait for a keypress
250   PRINT \'Press any key to continue...'
260   PAUSE
270 END DEFine carry_on
```

Listing Three

```
 10 REMark This program introduces WHEN ERRor.  If an error
 20 REMark occurs, the user is returned to that line number
 30 REMark to try again.
 40 REMark Procedures are heavily used to make flow of program
 50 REMark clearer.
 60 REMark
 70 initialise
 80 enter_num
 90 directory
100 PRINT \\'Program successfully completed.'
110 STOP
120 DEFine PROCedure initialise
130   MODE 0 : PAPER 0 : INK 7 : REMark Set Display
140 END DEFine initialise
150 DEFine PROCedure enter_num
160   WHEN ERRor
170     PRINT \'INVALID ENTRY!  (Must be a NUMBER)'
180     carry_on
190     CLS
200     RETRY : REMark Return to site of error.  Try again.
```

▶▶

```
210    END WHEN
220    CLS
230    INPUT 'Enter a Number:  ';num
240    PRINT \'Your number was ' & num
250    carry_on
260 END DEFine enter_number
270 DEFine PROCedure directory
280    WHEN ERRor
290      PRINT \'Microdrive ERROR.'
300      PRINT 'Place a program cartridge in MDV2'
310      carry_on
320      CLS : RETRY
330    END WHEN
340    DIR MDV2_
350 END DEFine directory
360 DEFine PROCedure carry_on
370    PRINT \'Press any key to continue...'
380    PAUSE
390 END DEFine carry_on
```

Listing Four

```
10 REMark This program allows the programmer to direct program
20 REMark execution from only one WHEN ERRor routine.  It
30 REMark accomplishes this by PEEKing the QL's BASIC VARIABLE
40 REMark area to discover the error number.
50 REMark
60 WHEN ERRor
70   REMark a little bell to alert the user
80   BEEP 2000,12 : PAUSE 5
90   BEEP 2000,6  : PAUSE 5
100  BEEP 2000,9  : PAUSE 5
110  errnum = PEEK_L(PEEK_L(163856) + 298)
120  SELect ON errnum
130    = -17
140      PRINT \'INVALID ENTRY!  (Must be a NUMBER)'
150      carry_on
160      at x,y : CLS 4 : CLS 2
170      REMark Reset cursor; clear screen
180      RETRY
190    = -7
200      PRINT \'ERROR at MDV2!'
210      PRINT 'Place a PROGRAM cartridge in MDV2'
220      carry_on
230      CLS : RETRY
240  END SELect
250 END WHEN : REMark Don't forget this line.  It's VITAL.
260 initialise
270 enter_num
280 directory
290 PRINT \\'Program successfully completed.'
300 PRINT 'You are now an expert error-trapper!'
310 STOP
320 DEFine PROCedure initialise
330   MODE 0 : PAPER 0 : INK 7
340 END DEFINE initialise
350 DEFine PROCedure enter_num
360   CLS
370   x = 0 : y = 16 : REMark variables to hold cursor position
380   PRINT 'Enter a Number: ';
390   INPUT num
400   PRINT \'Your number was ' & num
410   carry_on
420 END DEFine enter_num
430 DEFine PROCedure directory
440   CLS
450   PRINT 'Following is a directory of MDV2'\\
460   DIR MDV2_
470 END DEFine directory
480 DEFine PROCedure carry_on
490   PRINT \'Press any key to continue...'
500   PAUSE
510 END DEFine carry_on
```

trapping within the program against a user entering a string instead of a number. However, as we have no control over I/O errors (such as 'format failed'), we only include a statement to the user telling him how to re-enter the program.

Listing Three is our first introduction to **WHEN ERRor**. It will handle any type of error, and its only drawback is that we must include a **WHEN ERRor** routine at the beginning of each procedure. (I have included procedures here to demonstrate the clarity and modularity you gain from using these in your program.) Because the computer knows an error has occured (and which error it was), the **RETRY** command returns you to the specific line at which the mistake happened and re-executes from there.

Finally, in Listing Four we shall see how to PEEK the system variables on the QL to find the specific error code which was generated. By so doing, we can write just one error-trapping routine, one which will give a different result depending upon the code it receives. Finding the correct code works in this way: First, we must locate the base of the SuperBASIC area.

In the QL, this is not fixed, as it is in many other micros. Therefore, we find it by PEEKing memory location 28010h (163856 in decimal). The base of SuperBASIC is held as a 'long word', so we will use PEEK_L for this. The header associated with the SuperBASIC 'job' is 104 bytes (decimal) long, and we must add this to the start address. Last, we add the vector which points to the error code (BV_ERROR), which is 194 bytes (decimal) long, and PEEK the resulting address. This variable, too, is a 'long word', requiring us to again use PEEK_L. (Sounds complicated, doesn't it? But it isn't.) The formula for getting the error number is PEEK_L(PEEK_L(163856) + 298). Once we have this, it is quite simple to write a comprehensive block of code to trap our errors.

As a last note about **WHEN ERRor**, you should not think of it as providing copy protection for your programs. It does not recognise (CTRL)-(SPACE) (i.e. BREAK) as an error. Therefore, your program listings will be available to anyone who may be interested in them. However, if you force a program BREAK while an error is being processed, you may find yourself getting unpredictable (and unwanted) results at a later time.

Armed with an understanding of the QL's **WHEN ERRor** keyword and its syntax, you should now find that, while it is perfectly human to ERR, you will not be unduly punished for your mistakes.

CLOCK THIS!



Time to enter our competition and win yourself a Gremlin clock.

OK, take the flying ducks off the wall. We've got something much better for you, something that will become the aesthetic centrepiece of any front room and wouldn't look out of place in Fergie and Andy's new place (in fact, I'm told that they wanted one as a wedding present but these things aren't easy to come by).

What is it? What else but a high-tech, technicolour Gremlin clock! We've gotten together with the guys at Gremlin to offer five of these clocks as prizes to ZX readers, and for ten additional runners up there are copies of Gremlin's latest game, the mega-addictive Trailblazer, to be won.

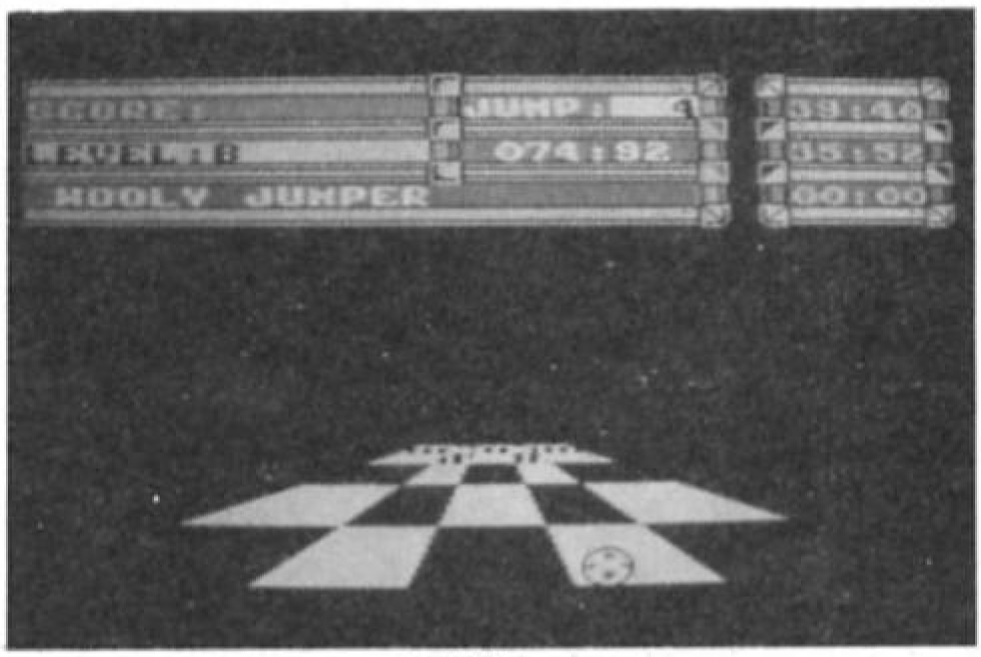
Anagrams

All you have to do to become the proud owner of one of these chronometric classics is to take the name of any Gremlin game, and see if you can come up with an interesting anagram made up out of that name. For instance, you could take Way of The Tiger and turn it into TOO FEW GET HAIRY (except that I cheated and made up an extra 'o', but you get the idea).

Just to get you started here are the names of a few Gremlin releases that you could use:
Monty on The Run
Jack The Nipper
Trailblazer
Way of The Tiger
Bounder

Once you've come up with an anagram, fill in the coupon on this page, and also write your anagram on the back of the envelope then send it to Gremlin Competition, ZX Computing Monthly, 1 Golden Square, London W1R 3AB.

Entries must arrive by first post on the 5th December 1985. This competition is open to all readers of ZX Computing Monthly, except employees of Argus Specialist Publications, Gremlin Graphics and Alabaster Passmore and Sons.



Trailblazer

Gremlin Competition

Game Title..........

Wacky anagram..........

Name..........

Address..........

....

....

Send your entry to Gremlin Competition, ZX Computing Monthly, 1 Golden Square, London W1R 3AB. Entries to arrive by 5th December 1986.

COMPETITION

MINDPLAY



Do you know the latest evil plan hatched from Barbaric Bryan's hideously warped mind? Not content with locking me in this dark, damp and dismal dungeon, not content with cold custard torture,, not content even with showing "The Price Is Right" on an out of reach video screen, 24 hours a day, he resorts to trying to ruin my macho image. I, of course, receive bulging bags of fan letters each day (I know they come, he just refuses to show them to me due to insane jealousy). The affront I am referring to is the "artwork" in the September issue. This portrayed my as having a concordian nose, and, in the drawing of a hand reaching out the grating, long, manicured and painted fingernails! But let my assure you reader, in reality I combine the best features of Mat Dillon and Richard Gere. Admittedly, the nails are longish, but I have been down here for three years and I certainly don't paint them. After all, how could I get to Boots? And if I did, they'd probably be right out of my shade of pink...

Snappy titles this month: HRH, Zzzz, and Moron (thankfully we did Journey To The Centre Of Eddie Smith's Head last month — it would have ruined the continuity). And, yonks after the Commodore version, Level 9's Jewels of Darkness has finally arrived for the Spectrum.

In last month's review of products from Interceptor's budget label Players, I said that far better than pushing out those poor games would be to re-relese some of the old Interceptor adventures. I'm glad to say that this is being done (not as a result of my comments, but who cares?). You can now buy Heroes Of Karn, Warlords and Jewels of Babylon amongst others for £1.99, a price they are ikeally suited to. This is because they are all marred by a poor vocabulary and pretty brief text. They do have some beautiful graphics though and fun problems, plus often a level of difficulty suitable for beginners, which is quite rare these days. They're minor classics and no adventurer's collection should be without at least one at such a price. If you cannot find them in the shops for so little, send for details from Interceptor at Mercury House, Calleva Park Industrial Estate, Aldermaston, Berks RG7 4QW.

News of adventure organisations continually comes my way. The Spectrum Adventure Exchange Club I mentioned in August is still going strong, and the same people now run Spectrum Adventurer, a tape magazine which costs two pounds. I looked at issue four and was quite impressed. There are reviews of less well known products, which tend to be ignored by most magazines; a well written help section; an entertaining column by none other than John Wilson (under a pseudonym); plus other bits and bobs. The magazine text is large and attractive. You also get part of a serialised adventure, which, although dealing with the hackneyed science fiction theme, seemed above average with some well written description (no graphics); certainly a keen adventurer could while away an enjoyable hour or two with it.

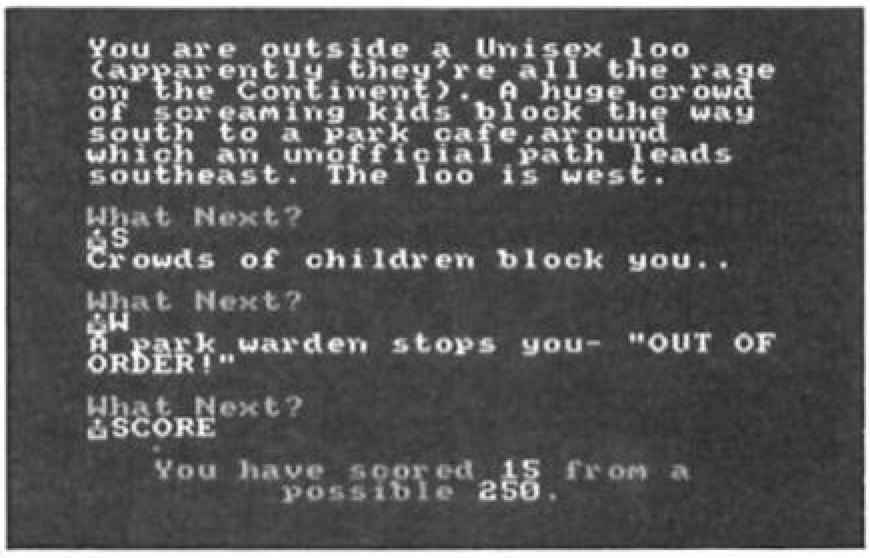
The main fault with the tape is that the magazine parts make little use of the computer medium. These is no index system and your are forced to flick through every page rather than choose a review, for example. A Ceefax style selection, or better still menu driven choices at the bottom of the screen, would greatly improve the product.

That said, Spectrum Adventurer is a good buy for the adventure nut; less devoted players won't find it as much fun. With enough reader support, it could become a very worthy mag indeed, and I hope they rectify the fault I mentioned (easily done). Write for more details to 4 Kilmartin Lane, Carluke, Lanarkshire ML8 5RT.

Roger Garret's Adventureline is apparently ending soon, though I haven't heard officially. Questionline is still going strong though, and all adventure clubs must watch out — because Tony Treadwell is starting a new, commercial adventure only magazine soon.

But enough of these deranged ramblings ... let the reviewing begin!

HRH
8th Day
£8.95



Whilst I'm not a great advocate of censorship, some of this game is of dubious taste and parents may feel that their children need protecting from its "adult" humour. But providing you are not easily offended, I think you'll love this. HRH is the most biting computer satire since Denis Through The Drinking Glass; this time the subject matter is (mostly) our royal family, and the irreverance is limitless.

You are horrified when your eagerly awaited DHSS envelope contains someone else's giro; then you notice the name and address: it belongs to Queen Elizabeth. And if you have hers — she must have yours! How will she afford the Queen Mother's meals on wheels on such a small sum? Will she have to sell the corgis to MacDonalds? What about her Oxfam clothing bill...? There's nothing for it but to return her giro personally.

You start in a post office, next to Princess Di waiting for her family allowance, and Prince William who is impersonating his father with a pair of huge rubber ears. Close by are further royals and before the end of the game you will have met the whole clan, on your travels around London. Their personalities will be familiar to all Spitting Image fans; drunken Margaret, Greek Phillip, stupid Edward, and randy Andy who provides many of those "adult" jokes I mentioned.

The humour itself is also along Spitting Image lines; some crude, some more subtle. I won't spoil the fun by giving too much away, but I particularly liked Madame Tussauds, where pride of place in both the Chamber Of Horrors and the Gallery Of Famous Murderers is taken by Margaret Thatcher. Later you face the gruesome task of impersonating her; yet when doing so, you won't be allowed into the palace!

Description is lengthy, entertaining and well written, and characters wander round in convincingly independent fashion. There are a few, fairly average split screen graphics.

An incredible amount has been packed into this Quilled game, not just text but many special messages and amazing complexity. While this is no fault, some of the commands — particularly EXAMINE — have had their responsiveness sacrificed in order to fit everything in. Despite this misgiving, HRH is technically very sound. Vocabulary is adequate, but at a few points too obscure. Coming from a small company, I was pleasantly surprised at the skillful presentation. The actual game has polished appearance, RAM SAVE/LOAD is included; and the packaging competes with many of the software big boys. My only major gripe is that this is a pound or two overpriced for a mail order product.

The actual adventure isn't quite up to Monster Hit standard. However, I had so much fun playing HRH, and its sheer originality and energy comes as such a welcome breath of fresh air in the face of countless middle earth games from Level 9 et al, that I've given one anyway. 8th Day are a company to watch

I sincerely hope they reach a distribution deal to get HRH into the shops, but in the meantime write to 18 Flaxhill, Moreton, Wirral, Merseyside L46 7UH.



Zzzz
Mastertronic
£1.99

You are trapped in your dream and must awake; though this game is more likely to have sent you to sleep in the first place. Dreaming could be the theme for an excellent adventure, but Zzzz fails because instead of capturing the weird nocturnal atmosphere or outrageous imagination of the best dreams, the authors have used the scenario as an excuse to fling together random and uninspired ideas. Also lacking is the warped logic which makes the wildest dreams feel continuous.

And Zzzz has many other faults, which is a shame because there are some good ideas. The best of these is the presence of icons replacing common commands (directions, GET/DROP, etc.), while text input is retained for rarer words. Unfortunately, a massive bug means these have a life of their own and automatically select random commands, even while you are typing, making the game virtually unplayable. Furthermore, the delete key won't function as it's used for icon selection. John Wilson tells me that these faults disappear if a joystick interface is plugged in, but this is not mentioned in the instructions; besides, there should be no need. The instructions are dreadful by the way, picturing non-existent icons — they were written for the CBM version.

The non-icon vocabulary is small, and you are required to enter ridiculously obscure commands like THUMB A LIFT. Examine is usually unresponsive. The text is reasonable but some of the graphics are pathetic. The whole thing is horribly unfriendly and not worth your time or money.





MORON
Atlantis
£1.99

Following the Marie Celeste earlier in the year, another science-fiction game from Atlantis, which also includes elements of time travel — killing two adventure cliches with one stone? Particularly stupid aliens called — you've guessed it — morons, stumble across the Pillars Of Time and promptly lose them in their spaceship. Said vessel is now on collision course with earth so you have a limited time to fulfill your task, which, naturally, is to collect the pillars and save the world.

Quilled and Patched, Moron offers split screen graphics at nearly all the numberous locations, quite a feat given the memory limitations. These are attractive enough but look incredibly similar, even though there are subtle differences. Then again, I suppose a large spaceship would appear somewhat monotonous.

To fit them all in, the text had to be sacrificed, or perhaps massacred would be more appropriate. There is not description at all, just a location title like "The Teleport Room". There are many objects lying around, which compensates a little, but purists should avoid. Vocabulary is strict but not awful.



I've said before that science-fiction games tend to bore me, and Moron has nothing new to offer. However, some of the problems here might while away the odd hour, though what I've seen strikes me as too easy for expert adventurers.

Moron is not a bad effort but strikes me as amazingly average, nothing new or notable. If there's nothing better around then this is no rip-off, but neither is it a must buy.



HELPLINE

Some help for David McCallum (not Illya Kuryakin himself, surely?), who's still stranded in sunny **Terrormolinos.** The correct hotel to go to is found by reading the brochure. Once in your room, drop your clothes, get and wear the trunks, then take, knot and wear the hanky. Suitably equipped with camera, snorkel and harpoon gun you'll be ready to start your holiday proper.

Meanwhile, back in Middle Earth, Michael Payne is troubled by trolls. The game is **The Hobbit,** naturellement. To kill the trolls and get the key, you must wait until dawn (i.e. keep typing WAIT until the program tell you it's light). However, you must do this when somewhere else, like the nearby path — otherwise it's boiled Michael on the trolls' menu.

In the decidedly more muddled world of **The Boggit,** the trolls are less easily dealt with — I suggest you listen very carefully to what the theologist says, then scour the washing powder shelves of your local supermarket. But before that, Stephen Page of Romford cannot escape his hole. He has found the diary (CLIMB CHEST for those who haven't) but cannot find the word combination to open the door. Turns out you were using the wrong birthday Steve: type Fordo's in!

Kris Daryani from Middlesex is struggling in an earlier satirical escapade of Fergus McNeil, **Bored Of The Rings.** He cannot find the poster of Rocky to give to the gay monsters. Well, starting by the pixie, enter the maze by going East. Then: E, N, E, E, S, W, S, E, GET POSTER, W, W, S, W. But don't hang around too long . . .

Mitchell Blower writes about **Sorceror Of Claymourge Castle.** "How do you find the towel and how do you enter the castle without the seed spell?" he asks, referring to hints in a previous issue. The moat answers both questions. From nearby, GO MOAT. Obviously you must HOLD BREATH, before you SWIM DOWN, TAKE TOWEL, SWIM DOWN, SWIM FAST, and SWIM UP. You will now be in the castle, and can save the seed spell for opening a stone door, later. The moat will need a further visit, with a deep treasure dive that can only be survived by escaping with the Bliss spell.

David Heron from Perston has an (old style) 128 and is stuck late on in **Never Ending Story.** "How can I reach the top of the Ivory Tower, I keep going round in circles". That's because you are in a maze, Dave. The route through is like this: from the bottom of the main stairs go U, E, E, NE, U, W and W, where you will find some completely useless treasure, then W, SE, U, E, E, E, U to find the door. Just be polite and the game is near finished. The cape you have found has no use. And in response to a steady stream of enquiries, it is not possible or necessary to enter the Ivory Tower in part one.

Some more information about **Denis Through The Drinking Glass.** When I gave some hints a couple of months back, I may have misled you into exitting by the window first. In fact you need to journey out the front door to start with. Make sure you are wearing your suit, and carrying the letter (which you must hand in at the Private Eye offices), the Cherry Blossom (for dealing with Norman Fowler) and your teeth (from the tumbler in the bathroom). The gnashers should be worn in time to SMILE at the photographers. Before you leave for the first time, collect all the items you can and leave them in your room, as Maggie will lock you in on your return. BUY TICKETS from the copper in the road then head NE.

Thanks as usual to the help of Rochdale's John Wilson.

Help The Helpline

The helpline has been running over six months now, and in that time I must admit (though not too loudly, otherwise Bryan will whip me even harder) that some problems have come in which even I cannot solve. Now it is an undeniable fact that ZX readers are the most intelligent beings in the entire universe, so I call on you to do a good deed and help your fellow creature. Please, please, please help. Please . . . (I think we get the gist Ed).

What I want you to do is fill in the "I can help solve" part of the coupon if you have completed any of the following games: **Eureka (Roman), Runes Of Zendos, Dragonworld, The Secret Of St. Brides, The Pawn, Smugglers Cove, The Moreby Jewels, Castle Colditz (K-Tel), Aksheron, Castle Blackstar** or **Return To Ithaca.** With a few exceptions those are rather obscure, but someone out there must have played them. If you have, just put the coupon in an envelope and I'll be in touch for help from you! Obviously an SAE is not needed in this case; and I'll be happy to give you any help in return.

Write to me

Whether you're stumped by Seabase Delta, harassed by Hunchback or bamboozled by The Boggit, we can help. Fill in the coupon and send it to: Mindplay Helpline, Deepest Dampest Darkest Dungeons, ZX Computing, No. 1 Golden Square, London W1R 3AB.

A few rules: British correspondents, please enclose a stamped, addressed envelope if you want a personal reply rather than wait for the magazine to come out. If you are writing from abroad, just enclose an envelope — I'll add the postage. I try to respond within two months but I can take longer (on the other hand, you might receive an immediate reply). I ONLY DEAL WITH ADVENTURES. Not arcade games; nor technical problems (write to Crosswires about those). I'm not too hot on adcventures these days (Gargoyle games included). Finally, please write the name of the game you're writing about on the back of the envelope.

Another month's missive must draw to a close, but fret not, for I'll still be here next issue. Or will Bryan succeed in his twisted plans to bump me off? Or will I succeed in my twisted plans to bump Bryan off? There's only one way to find out — read Mindplay next month!

Title: ....

Company: ....

Problem: ....

....

....

I can help solve: ....

Name: ....

Address: ....

....

VENTURESPEAK

Part 2: Alan Davis' adventure series looks at command analysis.

This month brings you the second part of the VENTURESPEAK machine code program, and we'll be taking a look at the gentle art of "parsing the input" — or "command analysis", if you prefer.

You're going to need the machine code keyboard routine that you saved last month, as we need to patch this month's program onto it to produce a single block of code. Listing 1 is the assembly language program for the "parser" — this is the program which will scan through a typed input, checking the words against a set of vocabulary data. As you can see, it's a fairly lengthy routine so an assembler will make error-free entry considerably easier — but if you don't have an assembler all is not lost, since I've given a decimal dump of the code in Listing 2. All you need to do is type in Listing 3, RUN it, and then enter the numbers from Listing 2 in order, including the checksums after every fifth byte. Take things slowly, checking the screen display against the listing as you go along, and all should be well. When you've finished, the program will save the code to tape for you as "PARSER" CODE 60400, 750.

Parsing

So far, so good. We now need to weld the keyboard and parser sections together as follows:

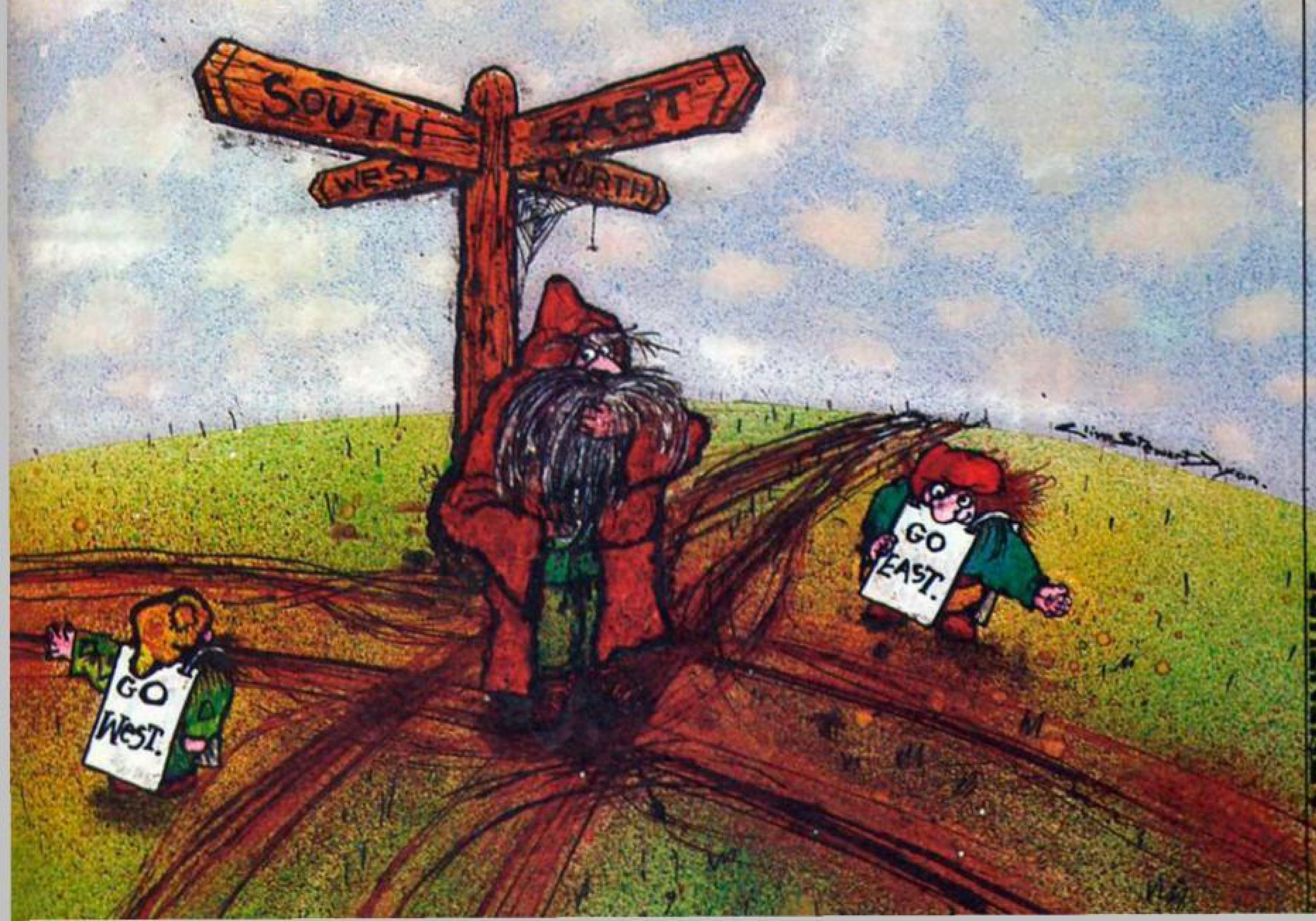
1) Reset the Spectrum using **RANDOMISE USR 0.**
2) Enter **CLEAR 59999** as a direct command.
3) Load in the keyboard code you saved last month **(LOAD "KEYBOARD" CODE).**
4) Load in this month's parser code **(LOAD "PARSER" CODE).**
5) Now save the whole lot as a single block with **SAVE "V-SPEAK" CODE 60000,1150** — and keep it safe somewhere until next month.

In addition to this single code block, it's probably wise to keep the two separate parts as well for the present, to facilitate checking in case you discover errors later.

Of course you'll be wanting to know just what this new routine does, and to test it out. We'll certainly be looking at how it works in this article, but as for testing it — well, I'm going to ask you to be patient and wait until next month. You see, the problem is that our parser is actually fairly useless at present, because it doesn't yet possess a vocabulary! (Rather like a chap who, though highly intelligent, has no background knowledge to draw upon . . .) To put vocabulary into the parser we'll need the VENTURESPEAK EDITOR — and that's our task for next month.

For the present though, I'm going to explain the basic principles underlying the parsing system so that when the time comes for you to use it, you'll be familiar with all the main features. We can't discuss this in a vacuum, so in Figure 1 I've given a very elementary vocabulary of words which you might expect to find in a typical adventure (printed out from the EDITOR program, in fact). Figure 1 itself raises a few points which need to be mentioned before we go any further:

1) Only the first three letters of any word are significant. (This can occasionally give rise to some confusion because the parser can't distinguish between words such as "RAVEN" or "RAVINE", although in practice I've never found this to be a serious problem, myself. My game "RUNESTONE" uses only



three-letter parsing).
2) Vocabulary is divided into three distinct types: VERBS, OBJECTS, and PEOPLE. This has advantages over the usual simpler subdivision into VERBS and NOUNS because it helps the error-trapping process when you write your adventure, as you'll see later.
3) Each item of vocabulary is assigned a number between 1 and 254 inclusive. Synonyms are catered for by assigning the same number to different words, so that the verbs SAY, TELL, ASK, and TALK, for example, are all assigned the number 1. This is a good place to make an **IMPORTANT NOTE:** generally you can assign any number you like to any verb you like, BUT IT IS ESSENTIAL THAT ALL VERBS IMPLYING SPEECH BE ASSIGNED THE NUMBER 1, if the parser routine is to work correctly.

Input

In practice what happens will go something like this. Your BASIC adventure program will invite input from the player by invoking the keyboard routine: LET M=USR 60000. The player then types in his command, presses ENTER when he's finished, and the routine returns to BASIC having stored the player's input as a string of bytes in the correct addresses above RAMTOP. We now want the parser to scan this stored command and decode it, so the next step is to call the parser routine with LET M=USR 60400. The parser now scans through the player's command, character by character, checking the words it finds against those in its vocabulary, ignoring any words it fails to recognise and skipping on to the next. It eventually stops, either when it reaches the end, or when it reaches a comma, full stop, or the word "AND" (these last items signifying that the player has typed in several commands at one go) and returns to BASIC.

The parser having done its job (almost instantaneously of course), we need to extract the fruits of its labours by PEEKing certain addresses. It makes for an easier life if the results of these PEEKs are assigned immediately to BASIC variables with suitably memory-jogging names, like this:

```
LET TELL=PEEK 61124
LET PERS=PEEK 61125
LET VB1=PEEK 61126
LET VB2=PEEK 61127
LET FK1=PEEK 61129
LET OB1=PEEK 61131
LET OB2=PEEK 61132
LET MORE=PEEK 61123
```

What this all amounts to is that you can extract from any one command the code numbers for up to three verbs, two objects,

Figure 1. Vocabulary list, with each verb, noun and object individually numbered.

```
VERBS              D     10        SHI    5
                   DOW   10        ARM    6
SAY    1           DES   10        BOO    7
TEL    1           GET   11        CHE    8
ASK    1           TAK   11        BOX    9
TAL    1           PIC   11        DOO   10
L      2           DRO   12
LOO    2           PUT   12
N      3           EXA   13        PEOPLE
NOR    3           REA   13
S      4           GIV   14        SAM    1
SOU    4           OFF   14        FRE    2
E      5           FIG   15        PET    3
EAS    5           KIL   15        JOH    4
W      6           ATT   15        JAC    5
WES    6           HIT   15        JOE    6
IN     7
ENT    7
OUT    8           OBJECTS
LEA    8
U      9           SWO    1
UP     9           DAG    2
CLI    9           SPE    3
ASC    9           KNI    4
```

Figure 2. Examples of command analysis.

A

```
EXAMINE THE DAGGER

 COMMAND ANALYSIS

 TELL:     0
 PERS:     0

 VB1 :     13
 VB2 :     0

 FK1 :     0

 OB1 :     2
 OB2 :     0
```

B

```
 SAY TO FRED
"EXAMINE THE DAGGER"

 COMMAND ANALYSIS

 TELL:     1
 PERS:     2

 VB1 :     13
 VB2 :     0

 FK1 :     0

 OB1 :     2
 OB2 :     0
```

C

```
PUT THE BOOK IN THE CHEST

 COMMAND ANALYSIS

 TELL:     0
 PERS:     0

 VB1 :     12
 VB2 :     7

 FK1 :     0

 OB1 :     7
 OB2 :     8
```

D

```
ASK PETER TO GIVE
THE KNIFE TO JOHN

 COMMAND ANALYSIS

 TELL:     1
 PERS:     3

 VB1 :     14
 VB2 :     0

 FK1 :     4

 OB1 :     4
 OB2 :     6
```

and two people — so that your BASIC program can then perform the necessary condition tests appropriate to your adventure. A few examples should make things clear.

Examining

Figure 2 gives examples of the analysis of a range of commands involving the basic vocabulary of Figure 1. (You might like to know that these figures are all screen dumps from the VENTURESPEAK EDITOR in "test" mode). Let's start with the first and simplest command — 2(a): "EXAMINE THE DAGGER". Only two words are really relevant here — the verb "EXAMINE", and the noun (in this case an object) "DAGGER". You'll see in the example that the parser has assigned the value 13 to VB1, 2 to OB1, and zero to everything else. Now look back at the vocabulary list in Figure 1, where you'll find that verb number 13 is "EXA" (for "examine"), and that object number 2 is "DAG" (for "dagger"). Makes sense? OK, then let's try something a little more complex.

The second example, Figure 2(b), shows the analysis of the command "SAY TO FRED "EXAMINE THE DAGGER" ", and we find VB1 and OB1 assigned just the same values as before (as we'd expect) but look: this time the variable TELL takes the value 1 (signifying speech) and the

Listing 1

Assembly language program.

```
*HISOFT GENS3M2 ASSEMBLER*
       ZX SPECTRUM

Copyright (C) HISOFT 1983,4
All rights reserved

Pass 1 errors: 00

              10 *D+
        20 *C-
        30 PARSER
60400   40          ORG   60400
60400   50 INIT     LD    HL,START
60403   60          LD    (NCOM),HL
60406   70          LD    (PTR),HL
60409   80          LD    A,(COUNT)
60412   90          LD    E,A
60413  100          LD    D,0
60415  110          ADD   HL,DE
60416  120          LD    A,255
60418  130          LD    (HL),A
60419  140          XOR   A
60420  150          LD    (TELL),A
60423  160          LD    (PERS),A
60426  170 RESTRT   XOR   A
60427  180          LD    (VB1),A
60430  190          LD    (VB2),A
60433  200          LD    (VB3),A
60436  210          LD    (FK1),A
60439  220          LD    (FK2),A
60442  230          LD    (OB1),A
60445  240          LD    (OB2),A
60448  250          CALL  COMMA
60451  260 LOOP8    LD    HL,(PTR)
60454  270          XOR   A
60455  280          LD    (EXTR),A
60458  290          CALL  FILTER
60461  300          LD    A,(EXTR)
60464  310          CP    1
60466  320          JP    Z,SORT
60469  330          LD    HL,(PTR)
60472  340          LD    A,(HL)
60473  350          CP    255
60475  360          JP    Z,SORT
60478  370          CP    44
60480  380          JP    Z,END
60483  390          CP    46
60485  400          JP    Z,END
60488  410          JP    LOOP8
60491  420 END      INC   HL
60492  430          LD    (PTR),HL
60495  440 SORT     LD    A,(VB1)
60498  450          CP    1
60500  460          CALL  Z,SWAP
60503  470          RET
60504  480 SWAP     LD    (TELL),A
60507  490          LD    A,(VB2)
60510  500          LD    (VB1),A
60513  510          LD    A,(VB3)
60516  520          LD    (VB2),A
60519  530          LD    A,(FK1)
60522  540          LD    (PERS),A
60525  550          LD    A,(FK2)
60528  560          LD    (FK1),A
60531  570          RET
60532  580 COMMA    LD    HL,(NCOM)
60535  590 LOOP     LD    A,(HL)
60536  600          CP    255
60538  610          JP    Z,ENDCOM
60541  620          CP    44
60543  630          JP    Z,SETCOM
60546  640          CP    46
60548  650          JP    Z,SETCOM
60551  660          INC   HL
60552  670          JP    LOOP
60555  680 SETCOM   INC   HL
60556  690          LD    (NCOM),HL
60559  700          LD    A,1
60561  710          LD    (MORE),A
60564  720          RET
60565  730 ENDCOM   XOR   A
60566  740          LD    (MORE),A
60569  750          RET
60570  760 FILTER   LD    A,(HL)
60571  770          CP    34
60573  780          CALL  Z,SKIPSP
60576  790          CP    32
60578  800          CALL  Z,SKIPSP
60581  810          CP    58
60583  820          CALL  Z,SKIPSP
60586  830          CP    255
60588  840          RET   Z
60589  850          CP    44
60591  860          RET   Z
60592  870          CP    46
60594  880          RET   Z
60595  890          INC   HL
60596  900          LD    A,(HL)
60597  910          CP    32
60599  920          JP    Z,SINGLE
60602  930          CP    58
60604  940          JP    Z,SINGLE
60607  950          CP    44
60609  960          JP    Z,SINGLE
60612  970          CP    46
60614  980          JP    Z,SINGLE
60617  990          CP    34
60619 1000          JP    Z,SINGLE
60622 1010          CP    255
60624 1020          JP    Z,SINGLE
60627 1030          INC   HL
60628 1040          LD    A,(HL)
60629 1050          CP    32
60631 1060          JP    Z,DOUBLE
60634 1070          CP    44
60636 1080          JP    Z,DOUBLE
60639 1090          CP    46
60641 1100          JP    Z,DOUBLE
60644 1110          CP    34
60646 1120          JP    Z,DOUBLE
60649 1130          CP    58
60651 1140          JP    Z,DOUBLE
60654 1150          CP    255
60656 1160          JP    Z,DOUBLE
60659 1170          JP    TRIPLE
60662 1180 SKIPSP   INC   HL
60663 1190          LD    A,(HL)
60664 1200          CP    32
60666 1210          JP    Z,SKIPSP
60669 1220          CP    34
60671 1230          JP    Z,SKIPSP
60674 1240          CP    58
60676 1250          JP    Z,SKIPSP
60679 1260          LD    (PTR),HL
60682 1270          RET
60683 1280 SINGLE   LD    HL,(PTR)
60686 1290          LD    A,(HL)
60687 1300          CALL  SETPTR
60690 1310          LD    DE,(VCV1)
60694 1320          LD    B,A
60695 1330 LOOP1    LD    A,(DE)
60696 1340          CP    255
60698 1350          RET   Z
60699 1360          LD    (TPNO),A
60702 1370          INC   DE
60703 1380          LD    A,(DE)
60704 1390          CP    B
60705 1400          JP    Z,VBFND
60708 1410          INC   DE
60709 1420          JP    LOOP1
60712 1430 DOUBLE   LD    HL,(PTR)
60715 1440          LD    A,(HL)
60716 1450          LD    DE,(VCV2)
60720 1460          LD    B,A
60721 1470          INC   HL
60722 1480          LD    A,(HL)
60723 1490          LD    C,A
60724 1500          CALL  SETPTR
60727 1510 LOOP3    LD    A,(DE)
60728 1520          CP    255
60730 1530          RET   Z
60731 1540          LD    (TPNO),A
60734 1550          INC   DE
60735 1560          LD    A,(DE)
60736 1570          LD    H,A
60737 1580          INC   DE
60738 1590          LD    A,(DE)
60739 1600          LD    L,A
60740 1610          LD    A,B
60741 1620          CP    H
60742 1630          JP    Z,FND1
60745 1640          INC   DE
60746 1650          JP    LOOP3
60749 1660 FND1     LD    A,C
60750 1670          CP    L
60751 1680          JP    Z,VBFND
60754 1690          INC   DE
60755 1700          JP    LOOP3
60758 1710 VBFND    LD    A,(VB1)
60761 1720          CP    0
60763 1730          JP    NZ,VB1DWN
60766 1740          LD    A,(TPNO)
60769 1750          LD    (VB1),A
60772 1760          RET
60773 1770 VB1DWN   LD    A,(VB2)
60776 1780          CP    0
60778 1790          JP    NZ,VB2DWN
60781 1800          LD    A,(TPNO)
60784 1810          LD    (VB2),A
60787 1820          RET
60788 1830 VB2DWN   LD    A,(VB3)
60791 1840          CP    0
60793 1850          RET   NZ
60794 1860          LD    A,(TPNO)
60797 1870          LD    (VB3),A
60800 1880          RET
60801 1890 SETPTR   INC   HL
60802 1900          PUSH  AF
60803 1910 LOOP2    LD    A,(HL)
60804 1920          CP    32
60806 1930          JP    Z,NEWWD
60809 1940          CP    34
60811 1950          JP    Z,NEWWD
60814 1960          CP    58
60816 1970          JP    Z,NEWWD
60819 1980          CP    255
60821 1990          JP    Z,LAST
60824 2000          CP    44
60826 2010          JP    Z,LAST
60829 2020          CP    46
60831 2030          JP    Z,LAST
60834 2040          INC   HL
60835 2050          JP    LOOP2
60838 2060 NEWWD    INC   HL
60839 2070 LAST     LD    (PTR),HL
60842 2080          POP   AF
60843 2090          RET
60844 2100 TRIPLE   LD    HL,(PTR)
60847 2110          LD    A,(HL)
60848 2120          CALL  AND
60851 2130          CP    1
60853 2140          RET   Z
60854 2150          LD    HL,(PTR)
60857 2160          LD    A,(HL)
60858 2170          LD    (CHR1),A
60861 2180          INC   HL
60862 2190          LD    A,(HL)
60863 2200          LD    (CHR2),A
60866 2210          INC   HL
60867 2220          LD    A,(HL)
60868 2230          LD    (CHR3),A
60871 2240          CALL  SETPTR
60874 2250          LD    DE,(VCV3)
60878 2260 LOOP5    LD    A,(DE)
60879 2270          CP    255
60881 2280          JP    Z,DWNVBS
60884 2290          CALL  SHORT1
60887 2300          JP    Z,VBMCH1
60890 2310          INC   DE
60891 2320          INC   DE
60892 2330          INC   DE
60893 2340          JP    LOOP5
60896 2350 VBMCH1   CALL  SHORT2
60899 2360          JP    Z,VBMCH2
60902 2370          INC   DE
60903 2380          INC   DE
60904 2390          JP    LOOP5
60907 2400 VBMCH2   CALL  SHORT3
60910 2410          JP    Z,VBFND
60913 2420          INC   DE
60914 2430          JP    LOOP5
60917 2440 DWNVBS   LD    DE,(VCOB)
60921 2450 LOOP6    LD    A,(DE)
60922 2460          CP    255
60924 2470          JP    Z,DWNOBJ
60927 2480          CALL  SHORT1
60930 2490          JP    Z,OBMCH1
60933 2500          INC   DE
60934 2510          INC   DE
60935 2520          INC   DE
60936 2530          JP    LOOP6
60939 2540 OBMCH1   CALL  SHORT2
60942 2550          JP    Z,OBMCH2
60945 2560          INC   DE
60946 2570          INC   DE
60947 2580          JP    LOOP6
60950 2590 OBMCH2   CALL  SHORT3
60953 2600          JP    Z,OBFND
60956 2610          INC   DE
60957 2620          JP    LOOP6
60960 2630 DWNOBJ   LD    DE,(VCFK)
60964 2640 LOOP7    LD    A,(DE)
60965 2650          CP    255
60967 2660          RET   Z
60968 2670          CALL  SHORT1
60971 2680          JP    Z,FKMCH1
60974 2690          INC   DE
60975 2700          INC   DE
60976 2710          INC   DE
60977 2720          JP    LOOP7
60980 2730 FKMCH1   CALL  SHORT2
60983 2740          JP    Z,FKMCH2
60986 2750          INC   DE
60987 2760          INC   DE
60988 2770          JP    LOOP7
60991 2780 FKMCH2   CALL  SHORT3
60994 2790          JP    Z,FKFND
60997 2800          INC   DE
60998 2810          JP    LOOP7
61001 2820 SHORT1   LD    (TPNO),A
61004 2830          INC   DE
61005 2840          LD    A,(DE)
61006 2850          LD    B,A
61007 2860          LD    A,(CHR1)
61010 2870          CP    B
61011 2880          RET
61012 2890 SHORT2   INC   DE
61013 2900          LD    A,(DE)
61014 2910          LD    B,A
61015 2920          LD    A,(CHR2)
61018 2930          CP    B
61019 2940          RET
61020 2950 SHORT3   INC   DE
61021 2960          LD    A,(DE)
61022 2970          LD    B,A
61023 2980          LD    A,(CHR3)
61026 2990          CP    B
61027 3000          RET
61028 3010 AND      CP    65
61030 3020          RET   NZ
61031 3030          INC   HL
61032 3040          LD    A,(HL)
61033 3050          CP    78
61035 3060          RET   NZ
61036 3070          INC   HL
61037 3080          LD    A,(HL)
61038 3090          CP    68
61040 3100          RET   NZ
61041 3110          CALL  SETCOM
61044 3120          LD    A,1
61046 3130          LD    (EXTR),A
61049 3140          DEC   HL
61050 3150          CALL  SETPTR
61053 3160          RET
61054 3170 OBFND    LD    A,(TPNO)
61057 3180          LD    B,A
61058 3190          LD    A,(OB1)
61061 3200          CP    B
61062 3210          RET   Z
61063 3220          CP    0
61065 3230          JP    NZ,OB1DWN
61068 3240          LD    A,B
61069 3250          LD    (OB1),A
61072 3260          RET
61073 3270 OB1DWN   LD    A,(OB2)
61076 3280          CP    B
61077 3290          RET   Z
61078 3300          CP    0
61080 3310          RET   NZ
61081 3320          LD    A,B
61082 3330          LD    (OB2),A
61085 3340          RET
61086 3350 FKFND    LD    A,(TPNO)
61089 3360          LD    B,A
61090 3370          LD    A,(FK1)
61093 3380          CP    B
61094 3390          RET   Z
61095 3400          CP    0
61097 3410          JP    NZ,FK1DWN
61100 3420          LD    A,B
61101 3430          LD    (FK1),A
61104 3440          RET
61105 3450 FK1DWN   LD    A,(FK2)
61108 3460          CP    B
61109 3470          RET   Z
61110 3480          CP    0
61112 3490          RET   NZ
61113 3500          LD    A,B
61114 3510          LD    (FK2),A
61117 3520          RET
60318 3530 START    EQU   60318
60315 3540 COUNT    EQU   60315
61118 3550 NCOM     DEFW  0
61120 3560 PTR      DEFW  0
61122 3570 TPNO     DEFB  0
61123 3580 MORE     DEFB  0
61124 3590 TELL     DEFB  0
61125 3600 PERS     DEFB  0
61126 3610 VB1      DEFB  0
61127 3620 VB2      DEFB  0
61128 3630 VB3      DEFB  0
61129 3640 FK1      DEFB  0
61130 3650 FK2      DEFB  0
61131 3660 OB1      DEFB  0
61132 3670 OB2      DEFB  0
61133 3680 VCV1     DEFW  0
61135 3690 VCV2     DEFW  0
61137 3700 VCV3     DEFW  0
61139 3710 VCOB     DEFW  0
61141 3720 VCFK     DEFW  0
61143 3730 CHR1     DEFB  0
61144 3740 CHR2     DEFB  0
61145 3750 CHR3     DEFB  0
61146 3760 EXTR     DEFB  0

Pass 2 errors: 00

AND     EE64     CHR1    EED7
CHR2    EED8     CHR3    EED9
COMMA   EC74     COUNT   EB9B
DOUBLE  ED28     DWNOBJ  EE20
DWNVBS  EDF5     END     EC4B
ENDCOM  EC95     EXTR    EEDA
FILTER  EC9A     FK1     EEC9
FK1DWN  EEB1     FK2     EECA
FKFND   EE9E     FKMCH1  EE34
FKMCH2  EE3F     FND1    ED4D
INIT    EBF0     LAST    EDA7
LOOP    EC77     LOOP1   ED17
LOOP2   ED83     LOOP3   ED37
LOOP5   EDCE     LOOP6   EDF9
LOOP7   EE24     LOOP8   EC23
MORE    EEC3     NCOM    EEBE
NEWWD   EDA6     OB1     EECB
OB1DWN  EE91     OB2     EECC
OBFND   EE7E     OBMCH1  EE0B
OBMCH2  EE16     PARSER  9A0A
PERS    EEC5     PTR     EEC0
RESTRT  EC8A     SETCOM  EC8B
SETPTR  ED81     SHORT1  EE49
SHORT2  EE54     SHORT3  EE5C
SINGLE  ED0B     SKIPSP  ECF6
SORT    EC4F     START   EB9E
SWAP    EC58     TELL    EEC4
TPNO    EEC2     TRIPLE  EDAC
VB1     EEC6     VB1DWN  ED65
VB2     EEC7     VB2DWN  ED74
VB3     EEC8     VBFND   ED56
VBMCH1  EDE0     VBMCH2  EDEB
VCFK    EED5     VCOB    EED3
VCV1    EECD     VCV2    EECF
VCV3    EED1

Table used:    826  from   850
```

variable PERS is given the value 2 (the number for Fred) corresponding to the person being addressed. Note that TELL is really only a flag which takes the value 1 or 0 according to whether speech is signified or not. It's the variable PERS which conveys the information about whom the player is trying to speak to.

So why do we need the extra variables on VB2 and OB2? Well, sometimes commands may involve *two* objects — and there may be instances when you want to distinguish between two slightly different types of action, such as putting objects ON or IN other subjects. Figure 2(c) shows the analysis of the command "PUT THE BOOK IN THE CHEST". See how the two verbs "PUT" (12) and "IN" (7) are picked up here, together with the two objects in order: "BOOK" (7) and "CHEST" (8).

The final example, 2(d) is of the type where speech to one person involves some kind of action with yet another person. And so the command "ASK PETER TO GIVE THE KNIFE TO JOHN" sets the speech flag (TELL) and sets PERS to 3 ("PETER"). As we'd expect, VB1 is 14 ("GIVE") and OB1 is 4 ("KNIFE") but in addition to this, FK1 has picked up the other person involved in the transaction ("JOHN"=4). We see here, by the way, how the separation of nouns into PEOPLE and OBJECTS eases the error checking process needed in every adventure. If a command is found to contain a verb such as "GIVE" or "FIGHT" (where some other person must of necessity be specified), it can be rejected as an unacceptable entry without further enquiry if FK1 is zero. Example 2(d) also illustrates the essential friendliness of VENTURESPEAK, particularly where speech commands are concerned. It matters not a jot what style of entry is used by the player — SAY TO PETER "GIVE JOHN THE KNIFE" for example, will be decoded with complete success by the parser.

I haven't mentioned how the system copes with multiple commands yet. Suppose the command GET THE SWORD AND GO NORTH has been entered. The parser will analyse the first part (GET THE SWORD) and then return to BASIC, but it will remember how far along the command it has scanned. If we LET MORE=PEEK 61123 (see above), the variable MORE will be assigned the value 1 — telling us in effect that another part of the command remains to be analysed. (For single commands MORE will be zero). This means we can arrange for the first command to be processed by an appropriate BASIC subroutine, and then test the variable MORE to see whether another analysis should follow, or whether a return to keyboard control is called for. However, the re-entry point to the parser routine is different should a continuation of analysis be necessary — on such occasions the parser is called with LET M=USR 60426. (This is because the usual entry at USR 60000 involves a tidying-up process which would cause the parser to "forget" the position it had reached on its previous scan).

A typical programming "flow chart", therefore, would be something like this:

(1) CALL KEYBOARD ROUTINE (USR 60000)

(2) CALL PARSER ROUTINE (USR 60400)

(3) PROCESS COMMAND IN BASIC

(4) "IF MORE THEN LET M=USR 60426": GOTO (3)

(5) GO BACK TO (1)

I hope that by now you've begun to get some idea, in principle, of what VENTURESPEAK can do for your adventure writing. If you're burning for some "hands-on" experience, don't worry. You'll be able to try out all these examples (and as many others as you wish) yourself with the help of the EDITOR program next time, when I'll also be offering more detailed explanations of how to incorporate the system into your own programs. Till then . . .

Listing 2

Decimal dump of parser machine code.

ADDR.	BYTES					CHECK SUM
60400	33	158	235	34	190	650
60405	238	34	192	238	58	760
60410	155	235	95	22	0	507
60415	25	62	255	119	175	636
60420	50	198	238	58	187	731
60425	238	175	50	198	238	899
60430	50	199	238	50	200	737
60435	238	58	201	230	50	777
60440	202	238	58	203	230	931
60445	50	204	238	205	116	813
60450	236	42	192	238	175	883
60455	50	218	238	205	154	865
60460	236	58	218	238	254	1004
60465	1	202	79	236	42	560
60470	192	238	126	254	255	1065
60475	205	79	236	254	44	818
60480	202	75	236	254	46	813
60485	202	75	236	195	35	743
60490	236	35	34	192	238	735
60495	58	198	238	254	1	749
60500	204	88	236	201	50	779
60505	198	238	50	199	238	929
60510	50	198	238	58	200	744
60515	230	58	199	230	50	763
60520	201	238	50	197	238	924
60525	58	202	238	50	201	749
60530	238	201	42	190	238	909
60535	126	254	255	202	149	986
60540	236	254	44	202	139	875
60545	236	254	46	202	139	877
60550	236	35	195	119	236	821
60555	35	34	190	238	62	559
60560	1	50	195	238	201	685
60565	175	50	195	238	201	859
60570	126	254	34	204	246	864
60575	236	254	32	204	246	972
60580	236	254	58	204	246	998
60585	236	254	255	200	254	1199
60590	44	200	254	46	200	744
60595	35	126	254	32	202	649
60600	11	237	254	58	202	762
60605	11	237	254	44	202	748
60610	11	237	254	46	202	750
60615	11	237	254	34	202	738
60620	11	237	254	255	202	959
60625	11	237	35	126	254	663
60630	32	202	40	237	254	765
60635	44	202	40	237	254	777
60640	46	202	40	237	254	779
60645	34	202	40	237	254	767
60650	58	202	40	237	254	791
60655	255	202	40	237	195	929
60660	172	237	35	126	254	824
60665	32	202	246	236	254	970
60670	34	202	246	236	254	972
60675	58	202	246	236	34	776
60680	192	238	201	42	192	865
60685	238	126	205	129	237	935
60690	237	91	205	238	71	842
60695	26	254	255	200	50	785
60700	194	238	19	26	184	661
60705	202	86	237	19	195	739
60710	23	237	42	192	238	732
60715	126	237	91	207	238	899
60720	71	35	126	79	205	516
60725	129	237	26	254	255	901
60730	200	50	194	238	19	701
60735	26	103	19	26	111	285
60740	120	188	202	77	237	824
60745	19	195	56	237	121	627
60750	189	202	86	237	19	733
60755	198	58	237	50	190	743
60760	238	254	0	194	101	787
60765	237	50	194	238	58	777
60770	198	238	201	58	199	894
60775	238	254	0	194	116	802
60780	237	58	194	238	50	777
60785	199	238	201	58	200	896
60790	238	254	0	192	58	742
60795	194	238	50	200	238	920
60800	201	35	245	126	254	861
60805	32	202	166	237	254	891
60810	34	202	166	237	254	893
60815	58	202	166	237	254	917
60820	255	202	167	237	254	1115
60825	44	202	167	237	254	904
60830	46	202	167	237	35	687
60835	195	131	237	35	34	632
60840	192	238	241	201	42	914
60845	192	238	126	205	100	861
60850	238	254	1	200	42	735
60855	192	238	126	50	215	821
60860	238	35	126	50	216	665
60865	238	35	126	50	217	666
60870	238	205	129	237	237	1046
60875	91	209	238	26	254	818
60880	255	202	245	237	205	1144
60885	73	238	202	224	237	974
60890	19	19	19	195	206	458
60895	237	205	84	238	202	966
60900	235	237	19	19	195	705
60905	206	237	205	92	238	978
60910	202	86	237	19	195	739
60915	206	237	237	91	211	982
60920	238	26	254	255	202	975
60925	32	238	205	73	238	786
60930	202	11	238	19	19	489
60935	19	195	249	237	205	905
60940	84	238	202	22	238	784
60945	19	19	195	249	237	719
60950	205	92	238	202	126	863
60955	238	19	195	249	237	938
60960	237	91	213	238	26	805
60965	254	255	200	205	73	987
60970	238	202	52	238	19	749
60975	19	19	195	36	238	507
60980	205	84	238	202	63	792
60985	238	19	19	195	36	507
60990	238	205	92	238	202	975
60995	158	238	19	195	36	646
61000	238	50	194	238	19	739
61005	26	71	50	215	238	600
61010	184	201	19	26	71	501
61015	58	216	238	184	201	897
61020	19	26	71	58	217	391
61025	238	184	201	254	65	942
61030	192	35	126	254	78	685
61035	192	35	126	254	68	675
61040	192	205	139	236	62	834
61045	1	58	218	238	43	558
61050	205	129	237	201	58	830
61055	194	238	71	58	203	764
61060	238	184	200	254	0	876
61065	194	145	238	120	50	747
61070	203	238	201	58	204	904
61075	238	184	200	254	0	876
61080	192	120	50	204	238	804
61085	201	58	194	238	71	762
61090	58	201	238	184	200	881
61095	254	0	194	177	238	863
61100	120	50	201	238	201	810
61105	58	202	238	184	200	882
61110	254	0	192	120	50	616
61115	202	238	201	0	0	641

Listing 3

Machine code loader program.

```
   1 REM ** M/C LOADER
   2 REM
   3 REM
   5 CLEAR 59999
  10 PRINT "ADDR.";TAB 13;"BYTES
";TAB 26;"CHECK"'TAB 28;"SUM"''
  20 FOR I=60400 TO 61115 STEP 5
  30 PRINT I;
  40 LET S=0
  50 LET N=0
  60 INPUT "NEXT BYTE?"'X
  70 POKE I+N,X: LET S=S+X: PRIN
T TAB 6+4*N;" " AND X<100;" " AN
D X<10;X;
  80 LET N=N+1: IF N<5 THEN GO T
O 60
  90 INPUT FLASH 1;"CHECKSUM FOR
 THIS LINE?"'C: PRINT TAB 27;" "
 AND S<10;" " AND S<100;" " AND
S<1000;S: IF C<>S THEN PRINT "ER
ROR! ENTER THE LINE AGAIN.": BEE
P 1,30: GO TO 30
 100 NEXT I
 110 SAVE "PARSER"CODE 60400,750
```

ADVENTURE

More from the last outpost of ZX81 users.

This month we have great pleasure in handing over the content of this page to two of our readers who wrote in with a couple of useful routines:

The first one is a brilliant addition for our giant REM project and we will be sending both contributors tokens of our appreciation . . .

Dear Sir,

I have sent you a contribution for your giant REM program. My machine code program will sort a specified character array into alphabetical order in a matter of seconds. eg. A 500 by 10 array in less than 2 seconds. This program is therefore very useful for filing programs.

My routine occupies 151 bytes and is relocatable if 2 addresses are altered. I have included a HEX loader to poke in the machine code and by changing line 100 to the required starting address, the program will automatically alter the two fixed addresses and so relocate the machine code routine.

The routine will only sort two dimensional character arrays and will produce an error code 2 if the variable has not been set up, or an error code 3 if the dimensions of the array are wrong the routine is called from basic by the command:

PRINT USR start address, X$

Where X$ represents any character array. It is more convenient to define the variable, SORT, to hold the starting address and so the routine can be called by:

PRINT USR SORT, X$

I have also written an example program to demonstrate my routine and this is listed from line 1000 onwards. To use this program simply change start address in line 1010 to the starting address of the routine.

Paul Kecskemety, Cheshunt, Herts.

Listing 1

```
   1 REM 151 CHARACTERSXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
  10 LET A$="210C0037368A23360DE
7CD1C11DA4B0D2323237EFE02207D234
E2346ED433F40"
  20 LET A$=A$+"234C234623ED433D
40ED4B3F40AF323C40E5E5ED5B3D4019
EBE1CDDC40DCF240"
  30 LET A$=A$+"EB0D70B120ECCE13A
3C40A7C0ED4B3F400BED433F4070B120
D0C9E5D5C5ED4B3D"
  40 LET A$=A$+"401A962007231300
70B120F5C1D1E1C9C5D5E5ED4B3D40ED
5B1440EDB0D1D5ED"
  50 LET A$=A$+"4B3D40EDB02A1440
ED4B3D40EDB0323C40E1D1C1C9CF02"
 100 LET START=16514
 110 FOR I=START TO START+150
 120 POKE I,16*CODE A$+CODE A$(2
)-476
 130 LET A$=A$(3 TO )
 140 NEXT I
 150 LET J=START+90
 160 POKE START+60,INT (J/256)
 170 POKE START+59,(J/256-INT (J
/256))*256
 180 LET J=START+112
 190 POKE START+63,INT (J/256)
 200 POKE START+62,(J/256-INT (J
/256))*256
 210 STOP
1000 REM EXAMPLE PROGRAM
1010 LET SORT=START ADDRESS
1020 DIM E$(10,15)
1030 FOR A=1 TO 10
1040 FOR B=1 TO 15
1050 LET E$(A,B)=CHR$ (INT (RND*
26)+38)
1060 PRINT E$(A,B)
1070 NEXT B
1080 PRINT
1090 NEXT A
1100 PRINT ,,"PRESS A KEY TO SOR
T"
1110 PAUSE 4E4
1120 PRINT USR SORT,E$
1130 CLS
1140 FOR A=1 TO 10
1150 PRINT E$(A)
1160 NEXT A
```

When we started this project we gave a machine code routine for creating a REM line, since then we have many new readers and so this tip should be useful for them.

Dear Sir,

In your July issue, Page 81, you give a program requiring 235 reserved spaces in a REM statement. I have seen programs requiring even more space than this and it can be very boring and slow creating the necessary REM.

The following little listing makes the job much easier and is worth keeping as a stock item on tape.

```
1 REM — (Followed by exactly 94 zeros)
9 REM
10 LET R = X + 2
20 POKE 16512, INT (R/256)
30 POKE 16511, R — 256* INT (R/256)
40 FOR N = 16514 TO 16511 + R
50 POKE N,28
60 NEXT N
```

It is quite easy to count the zeros in line 1 as they represent exactly 3 screen lines less 2 positions. 32*3=96. If you want to provide, say, 309 spaces then create 2 more REM lines as duplicates of line 1 by editing line 1 and changing its line number: i.e. 1 REM, 2 REM and 3 REM. Each of these will represent 100 spaces. Add 9 zeros to the REM in line 9. Replace the x in line 10 with 309. Run the program and wait a few seconds for the program to run through.

You will find that 1 REM will have disappeared. Delete the rest of the unwanted lines.

To provide say 295 spaces create the same 3 REM lines as explained above (=300 spaces) and then remove 5 zeros from 3 REM, this time replacing x with 295, then RUN.

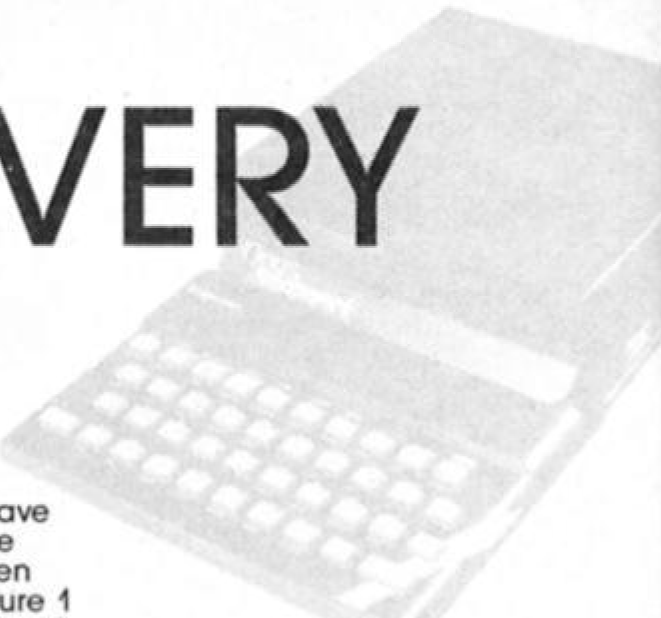
If zeros are not liked any other character can be used by changing the 28 in line 50 to the code of the desired character.

Yours faithfully,

L.G. Baumann, Cowies Hill, S.A.

PAGE 81

THE DISCOVERY COLUMN



John Wase with advice for Discovery Disc owners.

The Opus disc is notable for its reliability rather than speed in comparison with microdrives. If your principal use of the disc is for storing games, then reliability is probably the main criterion which you applied when you bought it. However, in contrast to a tape or a microdrive, a disc drive has the option of random access. In other words, if you specify a program out of fifty or so programs stored on the disc, then the disc operating system will immediately jump to the specified program. As you know, apart from the usual things one stores on tape, the disc system will also store lists of data, like telephone numbers. The Discovery unit is neat in the way it will set up special data files called Random Access Files: in this case it can jump to a specified item in the particular file. So if you just happen to be a tailor, and are storing the measurements of each of your customers' clothes on a separate file, the system allows you to pick out and load the file called "J.Jones" and then move immediately to, say, his arm or leg size.

Tracks and sectors

How is all this done? Well, the catalogue file occupies the first track on the disc, and the system always reads this first, comparing the information with that typed in (in our case, J.Jones). When it finds J.Jones on the catalogue file, it then reads information which tells it which track(s) and which sectors on those tracks are involved in storing the file. Then it knows which part of the disc to go to in order to find this information, which, of course, includes his arm or leg size.

Unfortunately, on the Discovery system, the catalogue file is only big enough to hold 110 records, and to make a living, you might want more than 110 customers. Well, the catalogue file can be expanded. I needed to do this for a specialised filing program, but unfortunatley the Opus instruction book was printed before the catalogue file-handling routines were finalised, and in some versions which include instructions, these actually don't work: if you have the later handbook, then the instructions have merely been omitted. The program in Figure 1 (which was obtained from Opus) should be used instead.

The listing

Having defined a function to simplify the reading of two-byte numbers (line 10) the program then asks you to input the drive number concerned and the number of blocks, or sectors (line 20). Line 30 opens the "CAT" channel to stream 0: stream 0 is a "funny" stream that doesn't always report errors. Line 40 then reads from the "CAT" channel and chickens out of more sectors are asked for than there is room. Finally, the new value is printed back to the "CAT" channel and the stream is closed (line 50).

Configuration of a Second Disc

Once you have got hold of the Discovery unit and become familiar with it, you might want to add a second disc unit with a power supply unit and the RAM chip, and you can get yourself almost a megabyte of storage in no time at all ...... except that when you come to FORMAT the thing, all you get is 178K on the screen when it is catalogued. This is because the internal software, which as emptied itself from the EPROM into the RAM chip hasn't been told that there is a disc of a different configuration in drive 2. There are two ways to tell it, for now that the information is in the RAM instead of the EPROM, it can be altered. One way is to read an appropriately configured disc in the drive, i.e. borrow one from a friend. The internal software detects the configuration, and is altered accordingly. The other is to use a program to do it. Here is a program, direct from Opus, which will do the trick (Figure 2).

This program is constructed on much the same lines as the last. Thus, line 10 defines the familiar function, as before. Line 20 opens stream 0 direct to the Opus RAM, using the "CODE" channel. Line 30 checks the version; the program won't work with the old "Spectrodos" system. Line 40 checks that the RAM has been properly fitted. Lines 50 and 60 use POINT to select an address, read a vector, adjust point and move to the new address, and repeat, so finding the location in which the information is to be stored. In line 70, you input the values for the number of sides (i.e. single or double sided), and the number of tracks (40 or 80). These are then put in the address which has been found (line 80), and the disc is formatted with the appropriate information in the title (line 90). If you have fitted your new disc as drive number 2, then change the "1" in line 90 to "2".

And...

If you have any programs or utilities for the Discovery disc, do send them for this column. We look forward to hearing from you.

Figure 1: Listing of the program to expand the catalogue file on the Opus system.

```
   5 REM program "Expand Cat"
  10 DEF FN a()=CODE INKEY$#0+25
6*CODE INKEY$#0
  20 INPUT "drive:",d'"no of blo
cks",b
  30 CLEAR #0: OPEN #0;" CAT ";d
 RND 16: POINT #0;2
  40 LET m=FN a()*0+FN a(): IF b
>m THEN CLEAR #0: OPEN #0;d;"x"
OUT 2^31: STOP
  50 POINT #0;1: RANDOMIZE FN a(
)*FN a()*0+b: PRINT #0;CHR$ PEEK
 23670;CHR$ PEEK 23671;: CLOSE #
0
```

Figure 2: Program to configure a disc

```
   5 REM program "Config"
  10 DEF FN a()=CODE INKEY$#0+25
6*CODE INKEY$#0
  20 CLEAR #0: OPEN #0;"CODE "
  30 IF USR 8<2.1 THEN PRINT "Wr
ong version": STOP
  40 IF USR 14070 THEN PRINT "Ra
m not fitted": STOP
  50 POINT #0;8192: POINT #0;FN
a()+10: POINT #0;FN a()+1
  60 POINT #0;FN a()
  70 INPUT "Tracks (40,80):",t'"
Sides (1,2):",s'"Insert new disk
 and press ENTER"; LINE a$
  80 PRINT #0;CHR$ t;CHR$ 18;CHR
$ (65+4*(USR 8<2.2)+16*(s=2));
  90 FORMAT 1;STR$ t+" tr,"+STR$
 s+" sd"
```

Jewels of Darkness

Peter Sweasey delivers the verdict on the revamped Middle Earth Trilogy.

Jewels of Darkness
Rainbird
£14.95

I very nearly didn't give this game a Monster Hit rating, but in the end I decided that even a below average Level 9 game is better than almost everything on the market. However, I'm extremely disappointed with Jewels Of Darkness; not so much for what it is, but because it isn't what it could so easily have been.

The games which form what was previously called the 'middle earth' trilogy are Colossal Cavern, Adventure Quest and Dungeon Adventure. The first is a version of the original mainframe game, with all the traditional puzzles, plus a special endgame where you rescue some elves in a race against time. Adventure Quest takes you, an apprentice magician, across varied terrain in your mission to destroy the Demon Lord. Dungeon Adventure is set just after said nasty's defeat, and sees you as an adventurer looting the treasure-filled caverns beneath his domain. The latter two have an obvious plot link but the first one "feels" very different and doesn't fit in particularly; the only common feature being the same starting point as AQ.

These are all well-crafted, enjoyable adventures, and no matter what I will shortly say this must be borne in mind. They remain impressive for their mass of locations and puzzles, the lengthy descriptions and varied responses (though not as wide-ranging as in Price Of Magik). In their original forms they have rightly become classics, and the versions in Jewels have been technically updated, though the actual games remain basically unchanged. They run faster and look (marginally) better. The sophisticated sentence constructions marvelled at in POM are mostly here too (like DROP ALL EXCEPT. . .). The marvelous type-ahead multi-tasking feature is provided. The originals lacked an EXAMINE command; this has been added, along with other vocabulary dimensions through the additional information that can be gained. The combination of three tapes provides months of entertainment and problems solving for a bargain price. The Jewels of Darkness are a joy to play.

Why then am I not lavishing the games with further praise? It would have been easy for me to go into auto-pilot for this review (Level 9 — must be brilliant), but when looked at closely it contains faults which I would criticise heavily in other games. Though these are minor shortcomings separately, taken together they spoil a potential masterpiece.

Presentation quality is poor. Descriptions, commands and responses are printed in the same yellow on black, with no gaps between them. Everything merges into an untidy mess, particularly if you play the full screen text version; yet this game above all others requires the player to read a tremendous amount. So why no use of a colour, or a more readable character set?

Then we have the 'graphics'. These are so stupefyingly



atrocious, so indescribably inept that I'd laugh if this wasn't a premier product. Just look at the lamentable screenshots we present here. I am not moaning at the inclusion of pictures — I realise they are necessary to gain wide distribution, and enjoy illustrations anyway — but these are so ghastly, no-one could want them. Have fewer of them, and make them mono-tone, if it means higher detail and the creation of something worth looking at. Instead we have blotchy, vague shapes and mindlessly inappropriate colouring (purple trees?). An illustrated version of Colossal offers fascinating possibilities which L9 have completley thrown away. Fortunately, a non-graphics version with expanded text, mostly longer EXAMINE responses, is provided on the other side of the tape (ignore the cassette label which claims that side is for 128 users only — a stupid, unnecessary error.)

The games lack RAMSAVE/LOAD, which is fast becoming standard, and should certainly be included in such a major release as this. It is claimed the commands couldn't be fitted in the Spectrum's memory, though many of the versions for other computers include them, and they are tantilisingly mentioned in the instructions. Even if memory was too tight in the graphics version — and seeing as they had no difficulty including the commands in Price Of Magik, this surprises me — why are they not in the text-only version, utilising some of the memory freed by the welcome disposal of the "graphics"? I for one would prefer a few messages not to be expanded in exchange for these useful features. A double punishment is that when you resort to conventional tape storage, which in games this size is frequently necessary, you are forced to use the dreaded Lenslok: EVERY TIME! Why not have just one code to crack at the start?

There are other, tiny niggles which all add up. There's no abbreviation for LOOK, unlike virtually every adventure on the market. You are only given a score rating on death; if you type SCORE or QUIT, you are just given a number. If the rating routine is there in memory, why not use it? There are no proper editing facilities, just delete: annoying during long sentences.

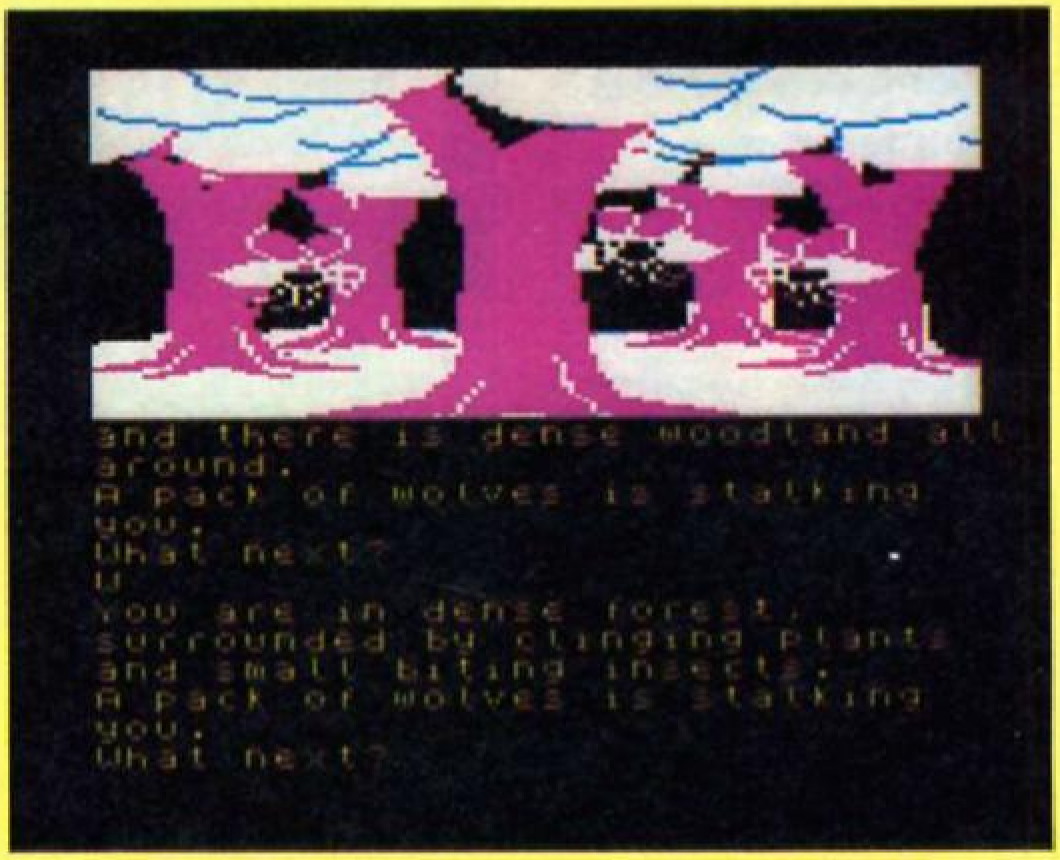
Plus, I think more could have been done to update the games themselves. Some independent characters perhaps? And Colossal looks rather hackneyed these days, so some new problems would have been welcome.

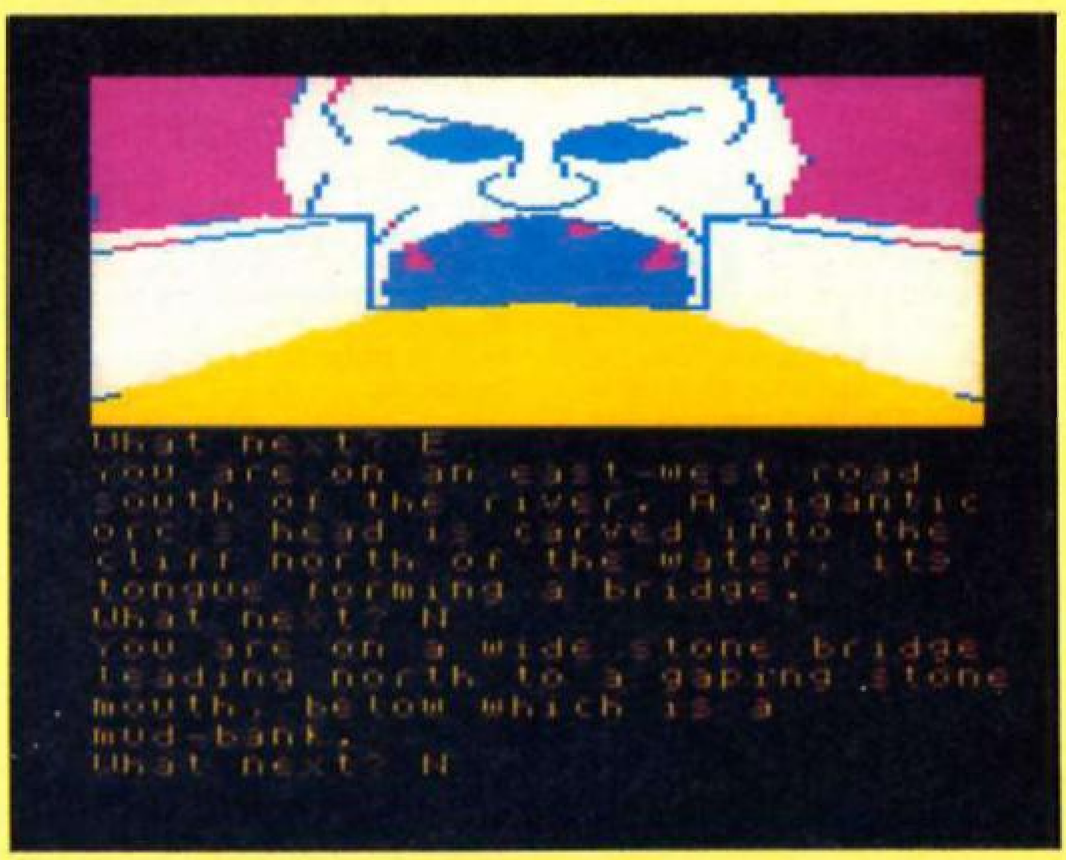
Virtually every fault I've outlined is easily corrected. But they're present, in what should be a particularly professional adventure, as a result of the Rainbird deal. Apart from the packaging, the BT tie-up seems to have had no effect. For example, there is no loading screen, just the same "Level 9" printed in different colours that has been used for several years. The adventures are good, so why not polish them to perfection? Nothing annoys me more than wasted opportunity.

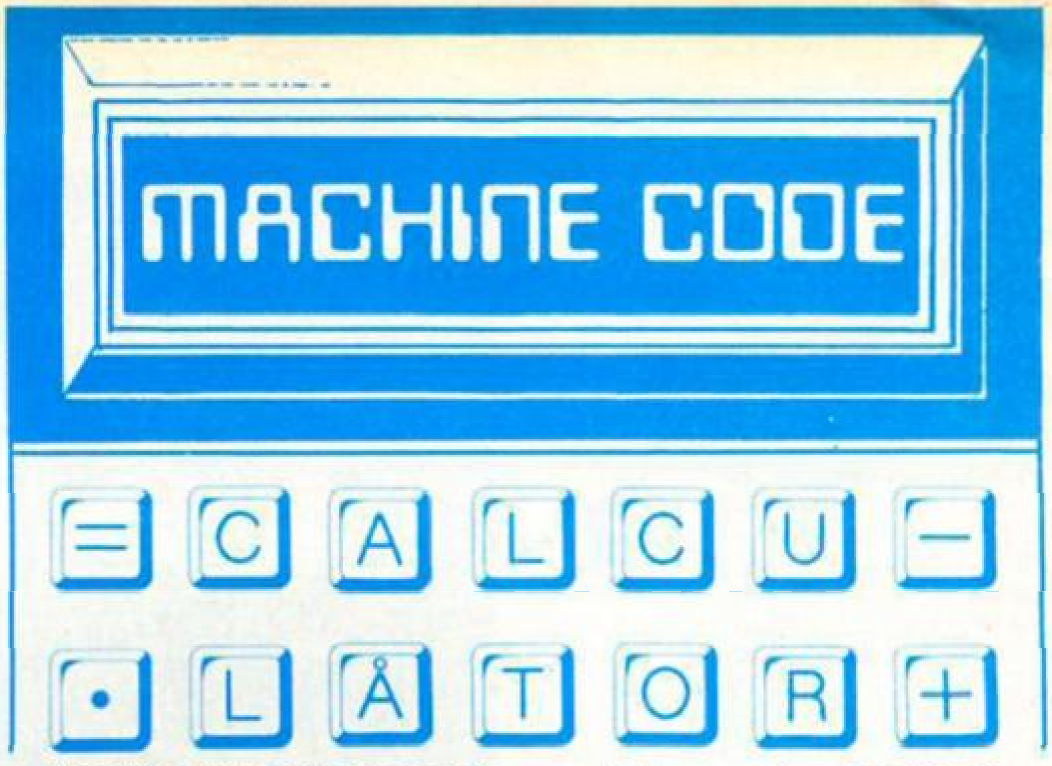
The attractive packaging is better than most by the way. The instructions are excellent; plus there's a highly readable, entertaining novella, which sets the scene for the latter two games. I expected this to be dull, standard fantasy fare, full of silly names, but it was unexpectedly humorous and gripping. Illustrations would have improved it though.

If you own more than one of these games already, then the revisions are probably insufficient to justify buying Jewels Of Darkness. If you have one or less, or if you are new to the world of adventuring or to the Spectrum, this package represents superb value for money and should keep you occupied for all the coming long, winter nights: perhaps the rest of 1987 as well!

MONSTER HIT







Toni Baker rounds off the series with a look at the function generator.

There is just one calculator instruction left to cover. It is the function "series", otherwise known as the *function generator*, or *series generator*. Its code will be a value between 80h and 9F. The last five bits of the code form a parameter, so that 86h means "series 6", 8C means "series twelve", 99 means "series twenty-five", and so forth. This is the single most powerful instruction in the whole of the calculator set. It is the function with which SIN and EXP and others were written in the ROM. With it we may create our own designer functions, or implement mathematical functions which are not present (and not otherwise possible) on the Spectrum. The "series" function is the calculator's final frontier. Essentially, what the "series" instruction does is to evaluate a polynomial expression. "Polynomial" is simply a highbrow piece of mathematical jargon — it means an expression which looks something like this . . .

$$a_1 + a_2 X + a 3 X^2 + ... + a_n X^{n-1}$$

I apologise to those of you not aquainted with mathematical notation. The same expression written in Spectrum BASIC would be as follows . . .

A(1) + A(2)*X + A(3)*X↑2 + ... + A(N)*X↑(N−1)

The dots in the middle of the above simply mean "and so on until". If you know the value of N you can fill in the missing chunk and put the whole lot into a single BASIC expression. If you don't know the value of N then you would have to use a FOR/NEXT loop from 1 to N in order to work it all out. But what does it all *mean?* Let's take it apart and find out.

Firstly the number 'n' — this is called the "degree" of the polynomial (technically the degree is n−1, not n). The "series" instruction specifies the number 'n' explicitly, so that, for example, "series 6" tells us that n equals six; "series 8" would mean that n equals eight. The parameter, which is part of the hex code (bits 4 to 0) actually specifies the degree of the polynomial. You should note that since you only have five bits in which to specify this parameter, its maximum value (other than zero) is hex 1F, or thirty-one. Note also that if you specify n to be zero, the ROM will mistakenly interpret this as two hundred and fifty six.

A polynomial expression such as we have been discussing consists of a number of terms, each separated by a "+" sign. The first term is a_1, and the last term is $a_i x^{n-1}$. In general, the *i*-th term will be $a_i x^{i-1}$. The value of the whole polynomial is therefore the sum of all of its terms. There are n such terms, and each of them contains the variable 'x' (except the first one — this is because in the first term x would have to be raised to the power of zero, and anything to the power of zero is one). Because every term except the first one contains the variable 'x' (and the first term is simply a number to be added), it follows that the whole thing is simply a function of x. In other words — you put in a value for x at one end, and you get a new number out at the other. This is in common with all of the other functions of the Spectrum.

The 'x' in this case is the value at the top of the calculator stack. When the "series" instruction is encountered, the value 'x' is removed from the calculator stack, the expression is evaluated, and the result put back onto the calculator stack in place of the original 'x'.

All we need to know now are the value of the 'a's. You can think of the 'a's as being a BASIC array A() dimensioned up to n, so that the first term is simply A(1), and the last term is A(N) multiplied by x↑(N−1).

Before we look at how to specify the value of the 'A's, let's have a look at how we may use such a polynomial to calculate functions.

Polynomials

Suppose that the value of 'x' is somewhere between minus one and plus one (ie that x is greater than minus one, and less than one) — we can of course ensure this since we already know how to manipulate the calculator stack. If x is zero then the value of the polynomial will simply be a , since all of the other terms will be multiplied by zero. If x is non-zero, but still between minus and plus one, then ABS(x^2) will be less than ABS(x): ABS(x^3) will be less than ABS(x^2), and so on. It follows, therefore, that if the 'A's are all roughly the same size, then each term will be smaller than the last. Furthermore, if the 'A's themselves also get smaller (ie if ABS(A(2)) is less than ABS(A(1)); ABS(A(3)) is less than ABS(A(2)); and so on) then each term will be smaller still than the last. Indeed it is possible to ensure that the last term is so small, by comparison to the first term, that if any more terms were added it would be outside the limits of the computer's accuracy, and would thus make no more difference. For instance, if A(1) equals 1, and if the last term equals 2^{-64} or less, then the Spectrum would have to round its answer up by simply ignoring the last term.

It is in theory possible to simulate any function whatsoever, provided that x is between minus one and plus one, that each term is suitably smaller than the last (although I can't go into the precise mathematical details of this condition in an article such as this), and that there are an infinite number of terms.

You see, with an infinite number of terms, we can make the polynomial closer and closer to the desired function with each new term. Unfortunately for us, we are only allowed a maximum of thirty one terms altogether. We can get round this problem by ensuring that the polynomial is close enough — it doesn't have to be *exactly* right. As long as the maximum error is smaller than the accuracy possible on the Spectrum, then the polynomial will calculate the function — at least to the limits of Spectrum accuracy.

So how do we work out the 'A's? — Well, mathematicians may care to use Taylor's or Maclaurin's Theorem (which I can't go into here), and everyone else will have to look the values up in books. Almost any A-level maths book (available from your local

library) will tell you what the correct series is to simulate SIN, or EXP, or whatever.

For instance, take the function 2↑X. If X is between minus one and plus one then the function can be simulated by the following polynomial. 2↑X is approximately equal to 1 + 0.69314718*X + 0.24022651*X↑2 + 0.055504109*X↑3 + 0.0096181291*X↑4 + 0.0013333558*X↑5 + 0.0001540353*X↑6 + 0.000015252734 *X↑7 ...

(Incidentally — I didn't work out the above numbers — I got them out of a book. I advise you to do the same. Mathematicians are, of course, welcome to work as much out for themselves as they want).

How does this help us? Well — you see we are still restricted to only using values of X between minus one and plus one (note: if you want to use negative numbers on the Spectrum for the above formula you'll have to use "X*X*X*X*X" instead of "X↑5", and so on). This is no good — the Spectrum can calculate EXP(X) for *all* values of X, not just small ones. To find the way round this problem we'll need to do a bit of maths. It's more difficult, but not, I hope, *too* difficult.

Firstly, note that EXP(X) is defined as e↑x, where e = 2.7182818

```
LET          u =x / (LN 2)
therefore:   x =u * (LN 2)
therefore: e↑x =e↑(u*(LN 2))
               =(e↑(LN 2))↑u
               =2↑u
```

Now, since u is a floating point number, it must have an integer part, and a fractional part. Thus:

```
LET           i =INT u
LET           f =u − i
therefore:    u =i + f
thus we     e↑x =2↑u
have            =2↑(i+f)
                =(2↑i) * (2↑f)
```

Now, since f is a fraction, between zero and one, we *can* use a polynomial to simulate the function. The polynomial I gave earlier may be used to calculate 2 f. All we need to do now is to multiply the result by 2 i (where i is an integer). We can do this simply by adding i to the exponent byte of the five-byte form of the number. This is the procedure used by the ROM to calculate EXP for all values of x.

Listing 2

```
10 CLS
20 FOR I = 1 TO 255
30 LET X = I/12.5
40 LET Y = LN FN G (X)
50 LET J = Y*4+10
60 IF J<176 THEN PLOT I,J
70 NEXT I
```

Well, I won't bore you to tears with any more maths, but I hope you can see how ingenious little tricks like the above can be used to ensure that polynomial approximations are only used on numbers between minus one and plus one. Such ingenious little tricks are necessary, because the polynomial formula won't work with numbers outside this range (for instance — the value of SIN(x) is always between −1 and +1, and yet, even if we used 256 terms, there would always be some large value of x for which $a_{256}\ x^{256}$ was greater than one. Clearly this would be the wrong answer. You *must* devise a means of ensuring that polynomial approximations to functions are always fed with numbers in range −1 to +1).

Instructions

Now — at last — we need to turn our attention to exactly *how* the values for the 'A's are passed to the "series" instruction. They are not specified explicitly — that would be too simple! Instead, we have to work out a new array, which I'll call B(), also dimensioned up to N. It is the values of the 'B's which are passed to the Spectrum. Therefore, our first task, knowing the value of the 'A's, is to work out the values of the 'B's. The BASIC program of Figure One will do this task for us. Don't worry too much about the algorithm used, just take my word for it — it works! (The mathematics required to prove that it works is beyond the scope of this article).

The program of Figure One doesn't print the 'B's in decimal — instead it prints them in the form required by the series instruction, converting them first to five byte form, and then to the compressed form used by the "stk-data" instruction. This form is also used by the "series" instruction.

Listing 1

```
10 LET K = 0
20 INPUT "N = ";N
30 PRINT "N = ";N
40 DIM A(N)
50 FOR I = 1 TO N
60 INPUT "A(";(I);") = ";A(I)
70 PRINT "A(";(I);") = ";A(I)
80 NEXT I
90 GO SUB 340
100 PRINT
110 PRINT FN H$(N+128)
120 FOR I = 1 TO N
130 LET K = B(I)
140 LET Q = PEEK 23627 + 256*PEEK 23628
150 FOR C = 3 TO 1 STEP −1
160 IF PEEK (Q+C+2) = 0 THEN NEXT C
170 LET D = PEEK (Q+1)
180 IF D<81 OR D>143 THEN GO TO 210
190 LET D = D − 80 + 64*C
200 GOTO 240
210 LET D = D − 80
220 IF D<0 THEN LET D = D + 256
230 PRINT FN H$(64*C);
240 PRINT FN H$(D);
250 FOR J = O TO C
260 PRINT FN H$(PEEK (Q+J+2));
270 NEXT J
280 PRINT
290 NEXT I
300 STOP
310 DEF FN H$(A) = FN K$(INT (A/16))
    + FN K$(A−16*INT (A/16)) + " "
320 DEF FN K$(A) = "0123456789ABCEDF"(A+1)
330 REM CALCULATES VALUES SUBROUTINE
340 DIM B(N)
350 FOR A = 1 TO N
360 FOR B = 1+N−A TO N STEP 2
370 LET X = B−1
380 GO SUB 520
390 LET J = Y
400 LET X = (A+B−N−1)/2
410 GO SUB 520
420 LET J = J/Y
430 LET X = (B−A+N−1)/2
440 GO SUB 520
450 LET J = J/Y
460 LET J = J*2↑(1−B)
470 LET B(A) = B(A) + J*A(B)
480 NEXT B
490 NEXT A
500 RETURN
510 REM LET Y = X FACTORIAL
520 LET Y = 1
530 FOR I = 1 TO X
540 LET Y = Y*I
550 NEXT I
560 RETURN
```

Gamma Function

```
BASIC
DEF FN G(X) = USR gamma_fn
The number following the keyword USR should be the address of the label GAMMA_FN
in the machine code program.

MACHINE CODE
EF          GAMMA    RST 28             x
C5                   store M5           (M5 contains x)
31                   duplicate          x,x
31                   duplicate          x,x,x
27                   int                x,x,INT x
03                   subtract           x,x-INT x
31                   duplicate          x,x-INT x,x-INT x
A2                   const half         x,x-INT x,x-INT x,1/2
03                   subtract           x,x-INT x,x-INT x-1/2
37                   gt zero            x,x-INT x,x-INT x> 1/2?
0066                 jump true,G_HALF   x,x-INT x
89                   series 9
E412EAAF03           .00014011071
```

▶ ▶ ▶

```
E7C62C2805          -.0015119361
EA0DCEC143          .0086552512
EC8C3556A3          -.034230555
ED5A25B37E          0.10651722
EF86344D75          -0.26211779
F018F3752D          0.59746487
F0EA14CE2B          -0.91437996
F17FDCCB17          1.9989256

                                        x,GAMMA(x+1-INT x)
01                  exchange            GAMMA(x+1-INT x),x
27                  int                 GAMMA(x+1-INT x),INT x
A1                  const one           GAMMA(x+1-INT x),INT x,1
03                  subtract            GAMMA(x+1-INT x),INT(x)-1
31                  duplicate           GAMMA(x+1-INT x),INT(x)-1,INT(x)-1
30                  eq zero             GAMMA(x+1-INT x),INT(x)-1,INT(x)=1
0017                jump true,G_1_TO_2  (Jump if INT(x) =1, ie if x is between
                                        one and two).

                                        GAMMA(x+1-INT x),INT(x)-1
31                  duplicate           GAMMA(x+1-INT x),INT(x)-1,INT(x)-1
36                  lt zero             GAMMA(x+1-INT x),INT(x)-1,INT x<1
0017                jump true,G_LT_1    (Jump if INT(x)<1, ie if x<1).
```

The following deals with the remaining case, ie if x is greater than or equal to two.

```
38                  end calc            GAMMA(x+1-INT x),INT(x)-1
CDD52D              CALL FP_TO_A        A:= INT(x)-1
47                  LD B,A              B:= INT(x)-1
EF                  RST 28              GAMMA(x+1-INT x)
A1                  const one           GAMMA(x+1-INT x),1
```

At each pass around the following loop, the topmost item on the calculator stack is assumed to be the "product-so-far", which I shall refer to as P_b. At any stage this will actually be the product x(x-1)(x-2)... to INT(x)-1-B terms. In addition, memory five will contain x-INT(x)+B+1.

```
                                        GAMMA(x+1-INT x),Pb
E5        G_LOOP_1  recall M5           GAMMA(x+1-INT x),Pb,x-INT(x)+B+1
A1                  const one           GAMMA(x+1-INT x),Pb,x-INT(x)+B+1,1
03                  subtract            GAMMA(x+1-INT x),Pb,x-INT(x)+B
C5                  store M5
04                  multiply            GAMMA(x+1-INT x),Pb-1
35FA                djnz G_LOOP_1       GAMMA(x+1-INT x),P0
```

At this stage the topmost item on the calculator stack is x(x-1)(x-2)(x-3)...K, where K is less than two (but greater than or equal to one).

```
04                  multiply            GAMMA(x)
38                  end calc
C9                  RET                 Return with correct value on calc stack.
```

The following applies for the case INT(x)=1. Note that since INT(x)=1 it follows that GAMMA(x+1-INT x) actually equals GAMMA(x), and that INT(x)-1 equals zero.

```
EF                  RST 28              GAMMA(x),0
02        G_1_TO_2  delete              GAMMA(x)
38                  end calc
C9                  RET                 Return with correct value on calc stack.
```

The following deals with the case where x is less than one.

```
EF                  RST 28              GAMMA(x+1-INT x),INT(x)-1
1B        G_LT_1    negate              GAMMA(x+1-INT x),1-INT x
38                  end calc
CDD52D              CALL FP_TO_A        A:= 1-INT x
```

▶ ▶ ▶

In other words, if you input the 'A's — which technically are the coefficients of a power series (don't worry if you don't understand the terms), then the program will miraculously transform the numbers into a new set of numbers — those used by the "series" instruction. The program leaves no work to you at all. The output of the program is the complete series instruction (in hex) required to evaluate the required polynomial, including the initial "series" byte. For extra clarity, you could, if you so desired, add an extra BASIC line at 285 PRINT,B(1) which would show you the 'B's in decimal as well as in hex.

If you now feed the bytes printed on the screen into a calculator program, then hey presto — your function will be operational!

Gamma

We shall demonstrate this procedure now by introducing a brand new function to Spectrum BASIC. It will be referred to in BASIC as FN G(), but the function will be defined mathematically as the GAMMA FUNCTION. It is very similar to the FACTORIAL function described last month, in that GAMMA(x) equals FACTORIAL(x—1) for all positive integer values of x, but whereas FACTORIAL only works for integers, GAMMA works for *all* numbers. A graph drawn of Y = GAMMA(x) will form a smooth continuous curve, and for all values of x it is true that GAMMA(x+1) equals x*GAMMA(x).

Furthermore, I shall go through, in detail, the process by which the function is being created. First of all we need a polynomial. I came across such a polynomial approximation in the book "Mathematical Methods in the Physical Sciences" by Merle C. Potter. The polynomial is only valid if x is between 0 and 1, but is more accurate if x is near zero. For this reason I have chosen only to use values of x between 0 and ½. The polynomial calculates the value of GAMMA(x+1) if x is in this range. The polynomial is as follows:

$$GAMMA(x+1) = 1 - 0.577191652x + 0.988205891x^2 - 0.897056937x^3 + 0.918206857x^4 - 0.756704078x^5 + 0.482199394x^6 - 0.193527818x^7 + 0.035868343x^8$$

Now, as I have said, I want to make sure that the polynomial is only used for values of x between 0 and ½, and yet I want the GAMMA function, FN G(), to work for all values of x, so we must find a way around this problem.

SPECTRUM PROGRAMMING

Firstly, we can make use of the rule GAMMA(x+1) equals x*GAMMA(x). We can use the rule repeatedly (in a loop) to reach any value of x, so long as the fractional part (the part to the right of the decimal point) is no greater than a half. For these remaining numbers we can use a different mathematical rule; for all x, GAMMA(x)*GAMMA(1−x) = PI/SIN(x*PI)

The first step, then, is to convert the polynomial into a "series" instruction. To do this we simply run the BASIC program of Figure One, and input 9 (because there are nine terms), and then input the nine numbers (1, −0.577191652, and so on). The program will print out the required form of the "series 9" instructions, which you will find incorporated into the main machine code program accompanying this article. The rest of it is just a hard slog, isolating the various cases, and dealing with them accordingly.

Incidentally, although I have used a loop to implement the GAMMA(x+1) = x*GAMMA(x) rule, this is not the only way of doing things. Another way would have been to call the GAMMA subroutine recursively (ie from within itself). I decided not to use this method because it would use up more memory on the calculator stack (an additional five bytes for each recursive call), but if you feel that that would not be too much of a disadvantage you might like to re-write the routine using this method.

Finally, the proof of the pudding is in the eating. The final test is whether or not it works. One way we can check is to calculate GAMMA(½) — the answer should come out to be SQR(PI). The best way to check, however, is to plot a graph. The function should produce a smooth continuous curve. The BASIC program of figure two will plot such a graph, and I will leave it to you to draw your own conclusions from it.

Well that's it from me for this series. We have now covered everything the calculator can do (or have we?). We have covered every single calculator instruction possible (or have we?). This now concludes our series. I hope you have enjoyed reading it. Good programming, and may the force be with you.

(But — just a thought — what would happen if you used "invalid" calculator codes? codes between 3E and 7F? With code 43h for instance, control could be directed into RAM, and enable the use of user-defined calculator instructions — defined in machine code of course. Interesting? Maybe one day the saga may continue . . . But not for the moment).

```
47                  LD B,A          B:= 1-INT x
EF                  RST 28          GAMMA(x+1-INT x)
A1                  const one       GAMMA(x+1-INT x),1
```

At each pass around the following loop, the topmost item on the calculator stack is assumed to be the "product-so-far", which I shall refer to as P_b. At any stage this will actually be the product (x+1)(x+2)(x+3)... to 1-INT(x)-B terms. In addition, memory five will contain x+1-INT(x)-B.

```
                                    GAMMA(x+1-INT x),P_b
E5      G_LOOP_2    recall M5       GAMMA(x+1-INT x),P_b,x+1-INT(x)-B
31                  duplicate       GAMMA(x+1-INT x),P_b,x+1-INT(x)-B,x+1-INT(x)-B
A1                  const one       GAMMA(x+1-INT x),P_b,x+1-INT(x)-B,x+1-INT(x)-B,1
OF                  add             GAMMA(x+1-INT x),P_b,x+1-INT(x)-B,x+2-INT(x)-B
C5                  store M5
02                  delete          GAMMA(x+1-INT x),P_b,x+1-INT(x)-B
04                  multiply        GAMMA(x+1-INT x),P_b-1
35F8                djnz G_LOOP_2   GAMMA(x+1-INT x),P_0
```

At this stage the topmost item on the calculator stack is (x+1)(x+2)(x+3)...K, where K is less than one, but greater than or equal to zero.

```
05                  divide          GAMMA(x)
38                  end calc
C9                  RET             Return with correct value on calc stack.
```

Finally, note that numbers such that x-INT(x) > 1/2 are dealt with separately, for greater accuracy. These are numbers such as -1.5 to -1; -0.5 to 0; 0.5 to 1; 1.5 to 2; and so on.

```
EF                  RST 28          x,x-INT x
02      G_HALF      delete          x
31                  duplicate       x,x
A1                  const one       x,x,1
01                  exchange        x,1,x
03                  subtract        x,1-x
38                  end calc        x,1-x
CD????              CALL GAMMA      Call the GAMMA subroutine recursively.
EF                  RST 28          x,GAMMA(1-x)
01                  exchange        GAMMA(1-x),x
A3                  const pi/2      GAMMA(1-x),x,PI/2
38                  end calc
34                  INC (HL)        GAMMA(1-x),x,PI
EF                  RST 28
C4                  store M4        (M4 contains PI)
04                  multiply        GAMMA(1-x),x*PI
1F                  sin             GAMMA(1-x),SIN(x*PI)
04                  mult            GAMMA(1-x)*SIN(x*PI)
E4                  recall M4       GAMMA(1-x)*SIN(x*PI),PI
01                  exchange        PI,GAMMA(1-x)*SIN(x*PI)
05                  divide          PI/(GAMMA(1-x)*SIN(x*PI))
38                  end calc        GAMMA(x)
C9                  RET
```

It can be shown algebraicly that PI/(GAMMA(1-x)*SIN(x*PI)) is equal to GAMMA(x). The proof of this is beyond the scope of this article, however I have made use of this fact to ensure maximum accuracy. It means that the "series" instruction is only ever passed values of x between zero and half. The last part of the program links the machine code to the BASIC DEF FN G statement.

```
2A0B5C  GAMMA_FN    LD HL,(DEFADD)  Point HL to user-define function record.
23                  INC HL
23                  INC HL          HL: points to 5-byte form of 'x'.
CDB433              CALL STACK_NUM  Place x on calculator stack.
CD????              CALL GAMMA      Calculate GAMMA(x).
C1                  POP BC          Delete STACK_BC address from machine stack.
C9                  RET
```

ZX COMPUTING

FULL CREDIT FACILITIES FOR ACCESS AND BARCLAYCARD.

Lineage: 46p per word. (VAT inclusive)

Semi display: £9.50 per single column centimetre + VAT. Ring for information on series bookings/discounts (minimum 2.5cm).

All advertisements in this section must be prepaid. Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request).

01 - 437 0699

Send your requirements to:
DUNCAN NEILSON
ASP Ltd., 1 Golden Square, London W1

SOFTWARE

WD Software

FOR THE QL:

JOSS — **base £13**

Forget that tedious, time-consuming syntax! Just move the cursor and press SPACE for all your file commands. Cursor keys or your joystick allow you to access microdrives (up to 8) and floppy discs (as many as our interface allows) with up to 150 files on each! Scroll and print directories, COPY, DELETE or PRINT any file, select TV or Monitor mode before LOADing or RUNning any program. You only use the keyboard to set the DATE or label a device when FORMATting. Easy to use with Psion and other software. No silly icons to learn - JOSS will TELL you what it's going to do! Programmer's toolkit and mass copying/printing utilities also supplied. Specify microdrive-only, Microperipheral or CST-compatible disc versions.

Ref QL7 — **base £7**

1300 useful QL references with ARCHIVE 2 search/print program. Too long for just one cartridge, so if you have RefQL5 just pay 2 and extra media cost to update.

Mdv Extension Cable — **£5.50**

Eight inches long, allows addition of extra microdrives to your QL. Twist it to put their slots facing you.

FOR THE QL: SPECTRUM; BBC; ELECTRON

WD Morse Tutor — **base £4**

Written to teach amateurs, now used by professionals too! Absolute beginner, or stretching your speed to 18 wpm, you won't find anything with more helpful features. What else can offer 100 random sentences as well as all the basics? Disc version unsuitable for BBC B+.

FOR THE SPECTRUM:

WorDfinder (Microdrive/disc only) — **base £8**

For CHEATING at crosswords. Finds m-ss-ng letters, solves, anagrms. 13,000 word vocabulary, so too long to share a cartridge. 10-letter word ending in ATE? No problem.

Tradewind — **base £3**

Sailing/trading strategy game with graphic surprises.

Jersey Quest — **base £3**

Text adventure in time. Background of Jersey folklore from Stone Age to Bergerac.

For export:

QL hardware and software from many sources. Ask for list/quotation.

ORDERING ADD COST OF MEDIUM, POSTAGE £1 OUTSIDE EUROPE.
Mdv or 5.25" floppy = £2 3.5" floppy = £4 Cassette = £0

Payment

By ACCESS/Eurocard/MasterCard or STERLING (UK bank cheques, Eurocheques, drafts or International GIRO). To:

WD Software (ZX), Hilltop, St. Mary, Jersery, C.I. tel: (0534) 81392

ZX 81 REVISITED! Factsheets, Software, A.Z.V.C. Club etc. Send SAE to: ACV, 1 Foxwell Square, Southfields, Northampton NN3 5AT.

PROOPS
TOTTENHAM COURT RD
ZX 81 LEARNING AID

A cassette based guide to programming the Sinclair ZX81 Microcomputer. It allows learning *on* the machine and *through* the machine. The package comprises a 160-page manual and 8 cassettes. There are 20 programs, each demonstrating a particular aspect of ZX81 programming, spread over 6 cassettes. The other 2 are blank practice cassetes.
PLEASE SEND **£4.50** (plus **£1.75** c/p)

PROOPS
BROTHERS LIMITED
52 Tottenham Court Road, London W1P 0BA
Telephone 01 636 4420

QL/SPECTRUM UTILITIES! Wide range of business/practical programs available. SAE/IRC for details (state Micro). S.D. Microsystems (ZX) PO Box 24, Hitchin, Herts.

ZX81/16K I CHING. Draws first three heragrams, prints results. Cheque £2.99 Eric Rosenbaum PO Box 2155 Jenkintown Pa, 19046 USA. Postpaid. Source Listing.

UTILITIES

SPECTRUM TAPE and DRIVE TRANSFER UTILITIES

TU8 — ESSENTIAL for ALL Spectrum 48K or Plus owners — our tape utility for backup and drive conversion

* Simple operation for fast loaders — accurate even if over 48K (tone and rate measurer).
* Integrated header reader, manages headerless blocks.
* Deals with clicky loaders — converts COMPLETELY to normal for tape load (plus easy drive transfer with TPM).

Plus many other featurers — a unique product!

COST only £6.99 on tape OR £7.99 on m/drive cart.

TRANSFER PACK 2 — FOR TAPE TO DRIVE TRANSFER

Package of programs including TU8, other utilities, allows drive transfer of many programs

* Details given on transfer of 40 programs.
* Manages virtually all clicky leaders. Bytes chopper/splitter, header reader etc.

State name of drive system (e.g. m/drive, Opus disc, etc).

ONLY £11.99 on tape OR £12.99 on m/drive cart.

INFORMATION SHEETS £1 each (up to No. 18 available) includes many top 20 hits — SAE for list. OVERSEAS: add £1 Europe, £2 others for each. UPDATE SERVICE: — SAE for details.

PROGRAMS CARRY OUT MONEY BACK GUARANTEE (not updates)

LERM, DEPT ZX, 11 BEACONSFIELD CLOSE, WHITELEY BAY, TYNE AND WEAR NE25 9UW.

EASTMON

Monitor/Disassembler

Disassemble and single step in RAM, ROM & IF1 with HARD COPY. Skip or execute CALLs. Hex and character entry. Block moves and fills. Clear manual - simple commands.

SPECTRUM 48 and SPECTRUM+
£4.99 inclusive.

MYLNHURST ELECTRONICS LTD
86 Eastern Way, Ponteland, NE20 9RE
Tel: Ponteland (0661) 24055

ACCESSORIES

SPECTRUM RESET SWITCH - Prevents damage to your computer when power plug pulled in and out. **Only £3.49 inclusive**. Cheque or P.O. to Central Electronics, St Andrews St, Greenock PA15 1HG. Other accessories. Write for free list.

ACCESSORIES

RGB OUTPUT ?
COMPOSITE VIDEO ?
GREEN SCREEN ?
TTL - ANALOGUE ?
COMP SYNC ?
BRIGHT ?

adapt electronics

CONFUSING ISN'T IT?

Talk of the problem solvers. We have the experience and expertise to advise on all aspects of computer display problems. eg:

★ **RGB OUTPUT MODULE** for Spectrum 16 or 48K. Much acclaimed — gives monitor quality pictures — no dot crawl. **£36.95**

★ **NEW** ★ **BRIGHT BOX** — Allows full use of the bright facility on analogue input monitors and monitor-TV's to produce full colour palette. Only for Spectrum 128. .... **£29.95**

Add P&P £1 (UK), £2 (EEC), £4 (Others)
Send SAE for our Free Monitor Fact Sheet

20 STARLING CLOSE, BUCKHURST HILL, ESSEX IG9 5TN. Tel: 01-504 2848

WARNING NOTICE

Advertisements placed in this magazine are to be in strict compliance with our standard conditions (copies of which conditions are available on request) and on the clear understanding that the advertiser warrants that his advertisement(s) does not infringe any copyright or condition of sale of any interested party in the advertised product.

Further, the advertiser indemnifies the proprietors of this magazine in respect of costs, damages, or any other claims brought against them as a result of legal action arising from the publication of the advertisement.

Any breach of these terms or the said conditions may result in prosecution of the advertiser by the proprietors.

Lineage rate: 46p per word (VAT inc.) Minimum **£7.20**.
Semi display: £9.50 per single column centimetre + VAT. Minimum size 2cm. No reimbursements for cancellations.
All ads must be pre-paid.
Write your advert in BLOCK CAPITALS in the grid below, ticking the section you wish it to appear under, INCLUDING YOUR NAME AND ADDRESS IN THE WORD COUNT and send it to: **ZX COMPUTING, ADVERTISEMENT DEPARTMENT, NO: 1 GOLDEN SQUARE, LONDON W1R 3AB.**

☐ REPAIRS ☐ SOFTWARE ☐ UTILITIES ☐ FOR SALE ☐ ACCESSORIES ☐ OTHERWISE PLEASE STATE

CLASSIFIED COUPON

ALL CLASSIFIED ADVERTISEMENTS MUST BE PRE-PAID. THERE ARE NO REIMBURSEMENTS FOR CANCELLATIONS.

I enclose my Cheque/Postal Order for £.......................for...... insertions, made payable to Argus Specialist Publications.
(Delete as necessary)

PLEASE DEBIT MY ACCESS/BARCLAYCARD NO

£.............FOR...........INSERTIONS EXPIRY DATE..............

Name ....

Address ....

.... POST CODE ....

DAYTIME TEL NO. ....

Signature .... Date ....

IF YOU DO NOT WISH TO CUT YOUR MAGAZINE, PHOTOCOPY THIS FORM

UTILITIES

Ray Elder examines a combined monitor/disassembler and a DIY poster maker.

EASTMON MONITOR/DISASSEMBLER
Eastway Systems
£4.99

This is an unusual and unexpected program, a "stand alone" Monitor/Disassembler. Usually these are produced as a support program for a company's assembler, and there is also the fact that there are already several on the market for the ageing Spectrum.

With this in mind the program must stand on whether it is easy to use and any special features it has.

Taking the latter first, Eastmon make a big thing about the fact that the DEFB's following a RST28 command are correctly interpreted (the 'use the calculator' command as Toni Baker's fans will know). It also corrrectly copes with the Interface 1 RST's.

You may or may not find this important, its useful to know that it'll cope though, especially if you use them a lot.

The program comes complete with a comprehensive, wordprocessed, dot matrix printed, fourteen page manual. It's all in there, but if your eyesight is not 20/20 then you may have problems with the tightly packed text.

Yep, it's easy to use and all the features that you are likely to need have been included, full printouts to any type of printer system and full details of the state of all registers at each step.

There is a choice of output, either a straightforward address, code, mnemonic print or a full frontal, everything exposed, printout in minute detail.

It may be late in arriving but it certainly does not suffer in comparison with established existing monitors. If you haven't got one and you spend a lot of time programming in machine code then this is highly recommended, it will certainly provide you with the information needed to track down that elusive bug; even in RST28 sections!

Contact: Mylnhurst Electronics Ltd, 86 Eastern Way, Darras Hall, Ponteland, Newcastle-upon-Tyne.



POSTER MACHINE
Softcat Micros
£5.95

Poster Machine is a versatile program which allows you to print screen pictures using most of the popular interfaces and a dot matrix printer.

This picture can be in one of eight predefined sizes from 3.25x2.5 inches to 69x60 feet, or in sections to produce really huge copies if that's not enough — though it is likely to be expensive in terms of printer ribbons!

Apart from working with most interfaces, ZXPRINT 3, KEMPSTON E etc (we tried it with them and it worked fine), the program provides options to invert the printed screen or add 'texture' in coloured sections to the copy.

You cannot create screens or pictures with it, these have to be generated by another graphics package such as their own excellent ANIMATOR1. In fact I would go so far as to suggest that it was developed as companion to this program and its more general application led them to market it separately.

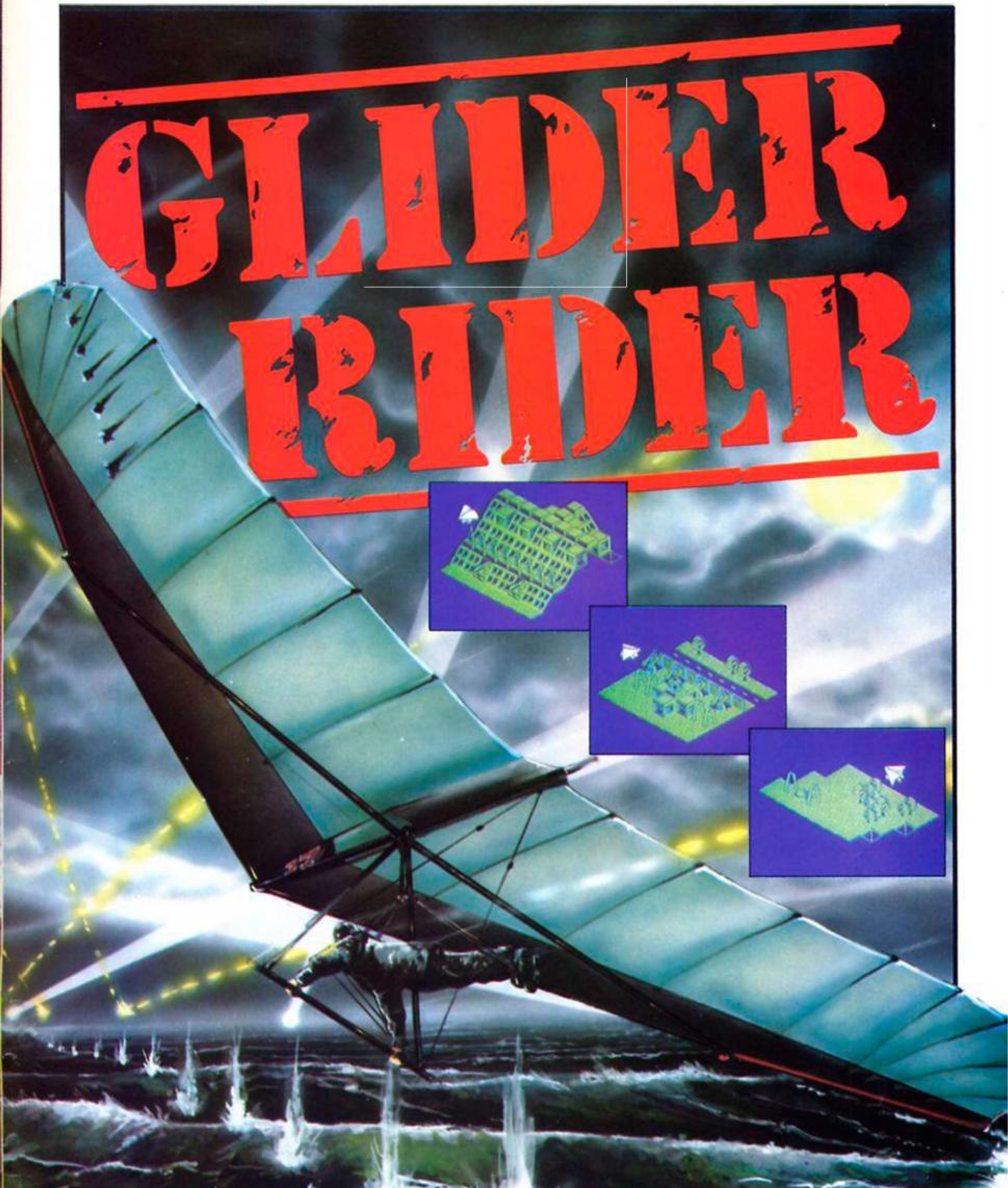
This is a simple and easy program to use and includes microdrive options, the only real difficulty came when fiddling with the expanded sections of a picture for the jumbo prints, but this isn't really a job you're likely to do too often anyway.

Really it would be more appropriately named Screen$ Printer, but I have no quibbles and if you need the specialist functions of flexible screen$ printing then this is the best I have seen (so far).

NB Poster Machine is being supplied free with Animator1 for a short time for £9.95, a good buy.







Trained to kill, you must penetrate the defenses of
Abraxas, destroy the terrorist H.Q. and glide to safety.

In case of difficulty, Glider Rider can be obtained from Quicksilva Ltd., Units 1 and 2, Conlon Developments, Watery Lane, Darwen, Lancs BBC3 2ET.

QUICKSILVA

Quicksilva Ltd., Victory House,
Leicester Place,
London WC2H 7NB.
Tel: 01-439 0666


A MISSION YOU CAN NEVER COMPLETE?
GALIVAN
GOLDEN COIN-OP
Hot Shot
POWER
POWER
Imagine
...the name of the game
SPECTRUM
£7.95
SPECTRUM
AMSTRAD
£8.95
COMMODORE 64
Imagine Software (1984) Limited
6 Central Street · Manchester M2 5NS
Tel: 061 834 3939 · Telex: 669977